NASA CONTRACTOR REPORT

NASA CR-2498

NASA CR-2498

A THEORETICAL METHOD FOR THE ANALYSIS AND DESIGN OF AXISYMMETRIC BODIES

T. D. Beatty

Prepared by
MCDONNELL DOUGLAS AIRCRAFT CORPORATION
Long Beach, Calif. 90801
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1975

1. Report No. NASA CR-2498	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A THEORETICAL METHOD FOR THE ANALYSIS AND DESIGN OF AXISYMMETRIC BODIES		5. Report Date MARCH 1975
		6. Performing Organization Code
7. Author(s) T. D. BEATTY		8. Performing Organization Report No.
		10. Work Unit No. 501-06-01-01-00
9. Performing Organization Name and Address McDONNELL DOUGLAS AIRCRAFT CORP. LONG BEACH, CALIF.		11. Contract or Grant No. NAS1-12986
		13. Type of Report and Period Covered CONTRACTOR REPORT
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION WASHINGTON, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

FINAL REPORT.

16. Abstract

A THEORETICAL METHOD IS PRESENTED FOR THE COMPUTATION OF THE FLOW FIELD ABOUT AN AXISYMMETRIC BODY OPERATING IN A VISCOUS, INCOMPRESSIBLE FLUID. A POTENTIAL FLOW METHOD IS USED TO DETERMINE THE INVISCID FLOW FIELD. THESE RESULTS YIELD THE BOUNDARY CONDITIONS FOR THE BOUNDARY LAYER SOLUTIONS. BOUNDARY LAYER EFFECTS IN THE FORCES OF DISPLACEMENT THICKNESS AND EMPIRICALLY MODELED SEPARATION STREAMLINES ARE ACCOUNTED FOR IN SUBSEQUENT POTENTIAL FLOW SOLUTIONS. THIS PROCEDURE IS REPEATED UNTIL THE SOLUTIONS CONVERGE. AN EMPIRICAL METHOD IS USED TO DETERMINE BASE DRAG ALLOWING CONFIGURATION DRAG TO BE COMPUTED.

17. Key Words (Suggested by Author(s)) AXISYMMETRIC BODY POTENTIAL FLOW; AXISYMMETRIC BODY BOUNDARY LAYER FLOW; COUPLED VISCOUS/INVISCOUS FLOW SOLUTIONS; FLOW SEPARATION ON AXISYMMETRIC BODIES	18. Distribution Statement UNCLASSIFIED - UNLIMITED STAR CATEGORY 34

19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of Pages 279	22. Price* $8.75

*For sale by the National Technical Information Service, Springfield, Virginia 22151

SUMMARY

A theoretical method is presented for the computation of the flow field about an axisymmetric body operating in a viscous incompressible fluid. This approach combines a smoothing routine, a potential flow method based on a surface source distribution, and a finite-difference boundary-layer method to accomplish the analysis. An empirical method used for modeling separated flow is shown to work reasonably well for cases of extreme flow separation. Results obtained by this method are presented which show very good agreement with experimental data. Suggestions are made for extending this method both to include a better model for separated flow and to calculate the "viscous" flow about axisymmetric bodies at angle of attack. A detailed instruction manual for inputing data to the computer program is given in Appendix A. Appendix B contains the necessary information to place this program on to a computer. This appendix also contains a complete description of output parameters from the computer program, as well as basic flow charts of some of the major subroutines. Appendix C contains a complete listing of the computer program for operation on either a CDC or an IBM computer.

TABLE OF CONTENTS

A THEORETICAL METHOD FOR THE ANALYSIS AND DESIGN OF AXISYMMETRIC BODIES

by T. D. Beatty
McDonnell Douglas Aircraft Corporation

One of the ultimate goals in aerodynamics is the achievement of the ability to obtain the real fluid flow field about an arbitrary three-dimensional configuration by theoretical calculation rather than by resorting to expensive and time consuming wind tunnel tests. The exact treatment of this problem requires the solution of the full Navier-Stokes equations, which is currently not practical. However, a good approximation to this real flow can be obtained by displacing the surface boundaries of the original body to account for viscosity as shown by Thwaites in Reference (1).

This technique of displacing the boundary surface to obtain a viscous solution has been used in two-dimensional flows quite successfully, as shown in References 2, 3, and 4. The extension of this approach to three-dimensional flow requires that appropriate computational routines be available to calculate the potential and viscous flow parameters. A potential flow routine which can calculate the flow about arbitrary three-dimensional bodies is available (Reference 5), although the comparable three-dimensional boundary layer method is not currently in the state of the art. At the present time, the general three-dimensional problem cannot be solved. However, both an axisymmetric potential flow method and an axisymmetric boundary layer method which can calculate the inviscid and viscous flow field about a body of revolution at zero degrees angle of attack are currently available.

Because of its simple nature and its common appearance in fluid dynamics, it was decided that a body of revolution would be a good starting point for the development of a three-dimensional method for calculating inviscid and viscous flow fields.

The axisymmetric potential flow routine (References 6 & 7), used in the present method was developed at the Douglas Aircraft Company under the guidance of A. M. O. Smith and has proven over the years to be an extremely versatile and accurate method, as well as the only purely axisymmetric potential flow method, generally available in industry today. This method has been well disseminated throughout industry; only a brief discussion will, therefore, be presented in a following section.

The boundary layer method presented in this report (Reference 8) is a finite difference technique which uses an eddy-viscosity concept to replace the Reynolds shear stress term. Since this method is relatively new and has been modified extensively since Reference 8 was reported, a detailed description will be presented.

The capability of the present method to determine the viscous flow about axisymmetric bodies is shown by correlations between the calculated results and experimental data.

Recommendations are presented for extending the present method to the calculation of the flow field about axisymmetric bodies at angle of attack.

DEFINITION OF SYMBOLS

A	Damping length or frontal area, wherever applicable
A^+	Damping constants
C_F	Total skin friction coefficient
C_p	Pressure coefficient
c	Chord
c_f	Local skin friction coefficient $\tau_w/(\frac{1}{2})\rho u_e^2$
D	Maximum diameter
f	Dimensionless stream function
G	Spot formation parameter
H	Shape factor, θ/δ
K_1	Mixing-length constant
k	Power to determine 2-D or axisymmetric flow
L	Reference body length
ℓ	Mixing length
P^+	Pressure gradient parameter
R_c	Chord Reynolds number, $u_\infty c/\nu$
R_D	Diameter Reynolds number, $u_\infty D/\nu$
R_x	Local Reynolds number. $u_e x/\nu$
R_θ	Momentum thickness Reynolds number, $U_e\theta/\nu$
r	Radial distance from axis of revolution
r_o	Local radius of body of revolution
T	Absolute temperature, °K or °R.
t	Transverse curvature term
U_∞	Free stream velocity
u_τ	Friction velocity, $\sqrt{\tau_w/\rho}$
u	x component of velocity

u_e	Velocity at edge of boundary layer
v	y-component of velocity
x	Distance along surface measured from leading edge or from stagnation point
y	Distance normal to the surface of the body

α	Angle between normal to the surface y and the radius r
$\propto$	Constant in outer eddy viscosity equation
β	Dimensionless velocity - gradient term, $\beta = (2\xi/u_e)(du_e/d\xi)$
γ_{Tr}	Transitional parameter
δ	Boundary layer thickness
δ^*	Boundary layer displacement thickness
ε	Eddy viscosity
ε^+	Ratio of eddy viscosity to kinematic viscosity, ε/ν
η	Transformed y-coordinate
θ	Momentum Thickness
μ	Dynamic viscosity
ν	Kinematic viscosity
ξ	Transformed x-coordinate
ρ	Density
τ	Shear stress

SUBSCRIPTS

c	Switching point between the inner and outer eddy viscosity formulas
e	Outer edge of boundary layer
i	Inner region
ℓ	Laminar

o	Outer Region
t	Turbulent
Tr	Transition
w	Wall
∞	Free-stream conditions

Primes denote differentiation with respect to η.

TECHNICAL DISCUSSION

Geometry Definition

The geometry input to the Douglas Neumann Potential Flow Program must satisfy two primary requirements: the coordinates must be distributed properly and the surface curvature must be smooth. These requirements are easily achieved on an analytical body shape, since the input coordinates may be calculated exactly for any prescribed distribution. However, some method of determining accurate input coordinates for an arbitrary axisymmetric body is necessary, since the body may not always be amenable to exact analytical definition. The approach adopted in the following method is to assume that the coordinates are input in the proper distribution about the body, but that they are not necessarily smooth. These two requirements will be discussed in some detail in the following sections.

Point distribution. - In order to obtain a high degree of accuracy in defining a pressure distribution when using the Douglas Neumann Potential Flow program, surface coordinates should be concentrated in regions of high surface curvature where rapid changes in the surface pressures would be expected. Since the total number of points per body is fixed, the distribution of these points about the body contour becomes extremely important. The Neumann program uses the input coordinates to create linear segments between points, thus approximating the body by a series of Frustums of Cones. The basis distribution required is then quite simple: more points and thus smaller segment sizes in regions of high curvature and less points and thus larger segment sizes in the other areas of the body. The basic guidelines to follow to insure proper point distribution are simply that the surface lengths of adjacent elements should not change by more than twenty to thirty percent and the maximum length of any segment should not exceed either five percent of the body chord or fifty percent of the local body thickness.

Smoothness of input coordinates. - The Douglas Neumann program, or any similar potential flow method, is sensitive to the derivative of the surface slopes, or the curvature of the surface. The surface defined by the input coordinates must therefore have smooth first and second derivatives. The approach used in the present method to smooth these coordinates, is a five point smoothing routine, which assumes that the input coordinates are smooth and continuous to graphical accuracy, i.e., points are chosen from a small graph (approximately a 10 inch chord). The output points from this routine will be moved very slightly to smooth the derivatives, but this movement will be negligible as far as the body shape is concerned. The equations used to accomplish this smoothing are as follows:

$$\bar{x}_j = \frac{1}{16}\left\{- x_{j-2} + 4x_{j-1} + 10x_j + 4x_{j+1} - x_{j+2}\right\} \tag{1a}$$

$$\bar{y}_j = \frac{1}{16}\left\{- y_{j-2} + 4y_{j-1} + 10y_j + 4y_{j+1} - y_{j+2}\right\} \tag{1b}$$

where x_j and y_j are the unsmoothed input coordinates

and $\bar{x}_j$ and $\bar{y}_j$ are the smoothed coordinates.

Potential Flow Method

The Douglas Neumann method, (References 6 and 7) is very general in that it can calculate the potential flow about virtually any body. There is no restriction, for example, to slender bodies; in fact, the "body" in question need not be a single body but may be an ensemble of bodies. In principle, the calculated solution may be made as accurate as desired by suitably refining the numerical procedure; accordingly, the so-called Neumann method is designated an exact method in this sense.

The Neumann method is based on the use of a distribution of source density over the body surface. Applying the condition of zero normal velocity on the body surface yields an integral equation for the source distribution. Specifically, the equation is a Fredholm integral equation of the second kind over the body surface. Once this has been solved for the source distribution, all flow quantities of interest, i.e., velocity, pressure, etc., can be calculated by rapid straightforward procedures. To implement this method on a computer, the body surface is approximated by a large number of small surface segments, over each of which the source density is assumed constant. The integral equation is replaced by a set of linear algebraic equations for the values of the source density on the segments. Input to the computer program consists of the coordinates of a set of points defining the body surface; these points are then used to determine the surface segments for approximating the body. There is no assumption made that the body can be analytically represented.

The usefulness of potential flow with its neglect of viscosity and compressibility is due to the fact that it is a good approximation to real flow under a wide variety of circumstances. With regard to viscosity, the program obtains useful results except in regions of catastrophic separation. To verify the usefulness of potential flow as a predictor of real flow, results calculated by the Neumann program have been compared with experimental data. Several collections of comparisons have been made. Reference 9 was a very complete collection but is now rather old. Reference 10 is a more recent collection that shows a smaller number of comparisons. In the calculation of the viscous flow about axisymmetric bodies it is necessary to add the boundary layer displacement thickness to the body as will be shown in a subsequent section. This results in an "open" trailing edge body. This "open" body can be evaluated by the Neumann program without any difficulty even though the boundary surface does not close. Reference 11 presents an explanation of this phenomenon which proceeds as follows: for a closed body the integral of the source density over the body is zero; for an "open trailing edge body, this integral is not zero, and a streamtube leaves the trailing edge of the open body which proceeds downstream and approaches infinity parallel to $\vec{u}_\infty$ as a constant cross section streamtube. Thus, the flow that is calculated may be thought of as that about a semi-infinite body consisting of the open body and an extension defined by this streamtube. The shape of the extension is unknown but is presumably unique, having both zero normal velocity and zero source density.

The potential flow program has many useful options available which do not pertain directly to the present development. The details of these options are described in References 12 through 17.

Boundary Layer Method

Basic boundary layer equations. - The calculation of the viscous flow over an axisymmetric body involves the solution of the laminar and turbulent flow equations. For laminar flows, the problem is strictly mathematical because the governing differential equations can be written exactly. For turbulent flows on the other hand, an exact solution of the governing equations is not possible. Consequently, in order to proceed at all, one must rely on a certain degree of empiricism. In the past, most of the work in this area has concentrated on so-called momentum and/or energy integral methods as a means of evaluating the viscous flow parameters. Thus, the exact mathematical solution to the problems of the turbulent flow was bypassed, leading to fast and simple methods with varying degrees of accuracy. These methods usually rely quite heavily on empirical correlations and generally are restricted to a limited range of flow conditions.

The Douglas Boundary-Layer Method (Reference 8), eliminates many of the disadvantages of the integral methods by proceeding to solve the full partial-differential equations governing the flow, thereby, being classified as a differential method. For two-dimensional and axisymmetric incompressible flows, turbulent boundary-layer equations contain terms involving time means of fluctuating velocity components known as Reynolds stress terms. At present the exact relationship between these terms and the mean velocity distribution in the boundary layer still remains unknown. In the present method, a relation based on the eddy-viscosity concept is used giving highly satisfactory results for a variety of flow conditions.

If the normal-stress terms are neglected, the incompressible turbulent boundary-layer equations for two-dimensional and axisymmetric flows can be written as in Reference 8:

Continuity

$$\frac{\partial}{\partial x}\left[r^k u\right] + \frac{\partial}{\partial y}\left[r^k v\right] = 0 \qquad (2)$$

Momentum

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = u_e \frac{du_e}{dx} + \frac{1}{\rho_\infty} \frac{1}{r^k} \frac{\partial}{\partial y} \left(r^k \tau\right) \qquad (3)$$

where

$$\tau = \tau_\ell + \tau_t$$

with

$$\tau_\ell = \mu_\infty \frac{\partial u}{\partial y} \qquad \text{(For laminar flow only)} \qquad (4)$$

$$\tau_t = - \rho_\infty \overline{u'v'} \qquad \text{(Additional term due to turbulent flow)}$$

and

$\overline{u'v'}$ = Reynolds shear stress term

$k = 0$ for two-dimensional flow

$k = 1$ for axisymmetric flow

The basic notation and coordinate scheme are shown in Figure 1, where U_∞ is a reference velocity and $u_e(x)$ is the velocity just outside the boundary layer. The coordinate system is a curvilinear one in which x is the distance along the surface measured from the stagnation point or leading edge, and y is measured normal to the surface. Within the boundary layer, the velocity components in the x- and y-directions are u and v, respectively. The body radius is r_o.

The boundary conditions for equation (3) are

$$u(x,0) = 0 \qquad (5a)$$

$$v(x,0) = 0 \qquad (5b)$$

$$\lim_{y\to\infty} u(x,y) = u_e(x) \qquad (5c)$$

Before equations (2) and (3) can be solved, they must be transformed to a coordinate system which removes the singularity at $x = 0$ and stretches the coordinate normal to the flow direction. First, these equations are placed in an almost two-dimensional form by the Probstein-Elliott transformation (Reference 18):

$$d\overline{x} = \left[\frac{r_o(x)}{L}\right]^{2k} dx \qquad (6)$$

$$d\overline{y} = \left[\frac{r(x,y)}{L}\right]^{k} dy \qquad (7)$$

where $r_o(x)$ is the body radius and $r(x,y)$ is a radius which accounts for the transverse curvature effect which will be subsequently discussed. A stream function Ψ is defined that satisfies the continuity equation (2):

$$\frac{\partial \Psi}{\partial y} = r^k u \qquad \frac{\partial \Psi}{\partial x} = - r^k v \quad , \quad \overline{\Psi} = \frac{\Psi}{L} \tag{8}$$

The resulting equations are transformed by the Levy-Lees transformation (Reference 19) in order to remove the singularity at $\overline{x} = 0$ and stretch the coordinates in the $\overline{x}$ and $\overline{y}$ directions. The Levy-Lees transformations are:

$$d\xi = \rho_\infty \mu_\infty u_e \, d\overline{x} \tag{9a}$$

$$d\eta = \frac{\rho_\infty u_\infty}{(2\xi)^{\frac{1}{2}}} \, d\overline{y} \tag{9b}$$

A dimensionless stream function, f, is introduced which is related to $\overline{\Psi}$ as follows:

$$\overline{\Psi} = (2\xi)^{\frac{1}{2}} f(\xi,\eta) \tag{10}$$

Combining the Levy-Lees and the Probstein-Elliott transformations given above we have

$$d\xi = \rho_\infty \mu_\infty u_e \left[\frac{r_o(x)}{L}\right]^{2k} dx \tag{11a}$$

$$d\eta = \frac{\rho_\infty u_e}{(2\xi)^{\frac{1}{2}}} \left[\frac{r(x,y)}{L}\right]^{k} dy \tag{11b}$$

Introducing an eddy viscosity term to account for the Reynolds shear stress terms,

$$\varepsilon \equiv - \frac{\overline{u'v'}}{\frac{\partial u}{\partial y}} \quad , \quad \varepsilon^+ \equiv \frac{\varepsilon}{U} \tag{12}$$

and a transverse curvature term t along with a pressure parameter term

$$\beta = \frac{2\xi}{u_e} \frac{du_e}{d\xi} \tag{13}$$

The momentum equation (3) then becomes, with $f' = u/u_e$,

$$\left[(1+t)^{2k} (1+\varepsilon^+) f''\right]' + ff'' + \beta\left[1-(f')^2\right] = 2\xi\left[f' \frac{\partial f'}{\partial \xi} - f'' \frac{\partial f}{\partial \xi}\right] \tag{14}$$

The boundary conditions given by equation (5) become

$$f(\xi,0) = f_w = 0 \tag{15a}$$

$$f'(\xi,0) = 0 \tag{15b}$$

$$\lim_{\eta \to \infty} f'(\xi,\eta) = 1 \tag{15c}$$

The momentum equation is then solved by a very efficient numerical scheme developed by Keller, (Reference 20) and applied to boundary layer calculations by Cebeci and Keller, (References 21 and 22).

Eddy viscosity equations. - The eddy viscosity concept is used to relate the time-mean fluctuating velocities to a mean velocity distribution as given in equation (12)

$$\varepsilon \equiv - \frac{\overline{u'v'}}{\frac{\partial u}{\partial y}} \tag{12}$$

A two-layer model of the eddy viscosity within the boundary layer will be used as shown in figure 2.

In the inner region of the boundary layer an eddy viscosity model, based on Prandtl's mixing-length theory, is used:

$$\varepsilon_i = \ell^2 \left|\frac{\partial u}{\partial y}\right| \tag{16}$$

where ℓ, the mixing length is given by

$$\ell = K_1 Y \tag{17}$$

A modified expression for ℓ has been developed by Van Driest (Reference 23) to account for the viscous sublayer close to the wall. This modification is

$$\ell = K_1 Y \left[1 - e^{-(Y/A)}\right] \tag{18}$$

where A is given by

$$A = A^+ \frac{\nu_\infty}{N} \left[\frac{\tau_w}{\rho_\infty}\right]^{-\frac{1}{2}} \tag{19}$$

and

$$A^+ = 26.0 \tag{20}$$

$$N = \left[1 - 11.8\ P^+\right]^{-\frac{1}{2}} \tag{21}$$

$$P^+ = \frac{\nu_\infty u_e}{u_\tau^3} \frac{du_e}{d\xi} \rho_\infty \mu_\infty u_e \left(\frac{r_o}{L}\right)^{2k} \tag{22}$$

$$u_\tau = \left(\frac{\tau_w}{\rho_\infty}\right) \tag{23}$$

Now for axisymmetric flows the value of ℓ is replaced by

$$\ell = .4r_o \ln\left(\frac{r}{r_o}\right)\left[1 - e^{-\frac{r_o}{A}\ln\left(\frac{r}{r_o}\right)}\right] \tag{24}$$

which is developed in reference 24. If transverse curvature effects are desired then

$$\frac{r}{r_o} = \frac{r_o + Y\cos\alpha}{r_o} = 1 + \frac{Y}{r_o}\cos\alpha$$

$$= 1 + t \tag{25}$$

$$\text{where } t = \frac{Y}{r_o}\cos\alpha$$

then ℓ becomes

$$\ell = .4r_o \ln(1+t)\left[1 - e^{-\frac{r_o}{A}\ln(1+t)}\right] \tag{26}$$

The eddy viscosity in the outer region of the boundary layer is given by

$$\varepsilon_o = \alpha\ u_e\ \delta_k^* \tag{27}$$

where δ_k^* is the boundary layer displacement thickness defined by

$$\delta^* = \int_0^{n_\infty} \left[1 - \left(\frac{u}{u_e}\right)\right] dn \tag{28}$$

which in the transformed plane becomes

$$\delta_k^* = \left[\frac{L}{r_o}\right]^k \frac{(2\xi)^{\frac{1}{2}}}{\rho_\infty u_e} \int_0^{n_\infty} (1-f')(1+t)^{-k} dn \tag{29}$$

where

$$1+t = \left[1 + \frac{2L \cos \alpha}{r_o^2} \frac{(2\xi)^{\frac{1}{2}}}{\rho_\infty u_e} \int_0^{n_\infty} dn\right]^{\frac{1}{2}}. \tag{30}$$

This relationship for ε_o is the same for two-dimensional or axisymmetric flows as shown in Reference 24.

Low Reynolds number effects. - The calculation of turbulent boundary layers about two-dimensional and axisymmetric bodies must often be done at low Reynolds number, i.e., momentum thickness Reynolds number, R_θ, less than 6000. Most of the boundary layer methods including the one presented above are based on empirical data which were obtained at high Reynolds numbers. A correction term to account for low Reynolds numbers which was developed by Cebeci (Reference 25) based on prior work by Coles (Reference 26) is, therefore, applied to the outer eddy viscosity by varying the α in equation (27) with R_θ in the following manner.

if $R_\theta < 425$ then $\alpha = (.0168)(1.55)$ (31a)

if $R_\theta > 6000$ then $\alpha = .0168$ (31b)

if $425 < R_\theta < 6000$ then $\alpha = .0168 \left[\frac{1.55}{1+\Pi}\right]$ where

$$\Pi = .55 \left[1 - e^{-.243\sqrt{\gamma} - .298\,\gamma}\right] \text{and} \quad \gamma = \left(r_\theta/425\right) - 1 \tag{31c}$$

Transverse curvature. - In developing the axisymmetric boundary layer equations a radius term is introduced as shown in equations (2) and (3). If the assumption is made that the body radius is very large compared to the boundary layer thickness then the radii in equations (2) and (3) reduce to the local body radius r_o and the effect of the transverse (i.e., circumferential) curvature in the momentum equation is neglected. If, however, the body radius is small compared to the boundary layer thickness then the effect of the transverse curvature cannot be ignored and r must be a function of the distance into the boundary layer, y. The relationship between y, r_o, and r is given by:

$$r = r_o + y \cos \alpha \tag{32}$$

As observed in figure 3, α is simply the surface slope in the longitudinal direction. i.e.,

$$\tan \alpha = \frac{dr}{dx} \tag{33}$$

For slender cylinders where $\alpha = 0°$,

$$r = r_o + Y \tag{34}$$

The inclusion of the transverse curvature terms in the boundary layer equations is shown in References 24 and 27 to substantially improve the accuracy of the calculation of the local skin friction as well as the other viscous parameters.

Transition region effect. - The boundary layer method has the capability of calculating transition from laminar flow to turbulent flow in two different ways. The first approach is to use the transition point as a switching point between laminar and turbulent boundary layer calculations. At the transition point the turbulent boundary layer calculations are started by activating the eddy viscosity coefficient. In general, especially at low Reynolds numbers this approach can lead to errors as shown by Cebeci in Reference 28. The second approach which is available uses the intermittancy factor given by Chen and Thyson (Reference 29) to modify the eddy viscosity equations to account for a region of transition. This modification was developed from the point of view of intermittent production of turbulent spots and is a further extension of Emmons' spot theory (Reference 30). The modification to be used is to multiply the inner and outer eddy viscosities equations (16) and (27) by the following parameter:

$$\gamma_{Tr} = 1 - e^{-Gr_o\left(x_{Tr}\right)\left[\int_{x_{Tr}}^{x} \frac{dx}{r_o}\right]\left[\int_{x_{Tr}}^{x} \frac{dx}{u_e}\right]} \qquad (35)$$

$$G = \left(\frac{3}{3600}\right)\left(\frac{u_e^{\,3}}{\nu_\infty^{\,2}}\right) R_{e_{Tr}}^{\;-1.34}$$

$$\text{where} \qquad R_{e_{Tr}} = \frac{u_e x_{Tr}}{\nu_\infty}$$

The effect of this transition region correction can be seen in figure 4 which compares experimental data to theoretical calculations for local skin friction with and without the above correction on a two-dimensional ellipse. This transitional effect will be assumed to be the same for axisymmetric bodies.

Boundary layer transition location. - The location of boundary layer transition from laminar to turbulent flow can be either input to the boundary layer method or calculated internally within the program. The approach used to calculate the transition location is one developed for two-dimensional flow by Michel (Reference 31) and later verified by Smith (Reference 32). This method correlates the local momentum thickness Reynolds number, R_θ and the local distance Reynolds number, R_x, as shown in figure 5 which comes from Reference 32. The procedure used is to calculate the values of R_x and R_θ at each station and to compare them to the curve in figure 5. If the value of R_θ is less than the value of $R_{\theta_{TR}}$ than transition has not been reached but if the value of R_θ is greater than $R_{\theta_{TR}}$ then transition has occurred.

The above method was extended to axisymmetric flow by the use of Mangler's transformation. The parameters R_θ and R_x are calculated by the axisymmetric boundary layer routine and they are then transformed to two-dimensional values by the following relationships:

$$\theta_{2-D} = \left(\frac{r_o}{L}\right) \theta_{AXISYMMETRIC} \qquad (36a)$$

$$X_{2-D} = \int_0^{x_{LOCAL}} \left(\frac{r_o}{L}\right)^2 (dX)_{AXISYMMETRIC} \qquad (36b)$$

These values of θ_{2-D} and X_{2-D} are used to determine values of R_θ and R_x which can be used in conjunction with figure 5.

A study of transition location calculation for axisymmetric bodies was recently completed by Kaups (Reference 33). In this study empirical methods due to Granville, Hall and Gibbons, and the method of Michel presented above were compared to the stability analysis technique of Smith (Reference 32). It was determined that for flows where transition occurred in an adverse pressure gradient all of the above techniques predicted transition fairly accurately. For flows where transition occurred in favorable pressure gradients, only the method of Smith (Reference 32) gave satisfactory results as shown in figure 6 which is taken from Reference 33. The method of Smith, however, requires extremely lengthy computer calculation times which makes it undesirable for the iterative type of calculation presented in this report. Therefore, based on the results of Reference 33, the method of predicting transition in the present program should not be used for flows with very large Reynolds numbers where the transition location might occur in a favorable gradient, but rather the transition point should be input to the program.

Calculation Procedure

The viscous flow field about an axisymmetric body is simulated by calculating the inviscid flow about an equivalent "viscous" body which is formed by adding the boundary layer displacement thickness to the original body surface. This technique of defining the inviscid body has been used quite successfully for two-dimensional flows as shown in Reference 2 and has also been used for axisymmetric flows as presented in Reference 34. This equivalent body is formed by combining the previously discussed geometry routine, potential flow method, and boundary layer method under control of the axisymmetric design and analysis method computer program known as ADAM.

Given the desired axisymmetric configuration and flow conditions, the ADAM program utilizes these sections, as shown in figure 7, in the following iterative manner:

1. Precise geometry definition for input into the potential flow program.

2. Calculation of the exact nonlinear potential flow for specified geometry and flow conditions.
3. Calculation of the viscous flow characteristics based on the results of the potential flow program.
4. Addition of boundary-layer displacement thickness to the basic geometry for each element.
5. Recalculation of the pressure distribution utilizing the potential flow program, based on the redefined geometry.
6. Recalculation of viscous flow field based on recalculated pressure distribution from redefined geometry, if desired.
7. Possible iteration of the above scheme; the degree to which this is required is presented in the subsequent discussion on correlations with experimental data.

The above technique must be modified when the boundary layer separates or when the local body radius approaches zero at the trailing edge of the body. When the dimension of the local body radius approaches zero at the trailing edge, the boundary layer equations become invalid since the $1/r$ term in equation (3) approaches infinity. When this occurs, the boundary layer results are ignored from this point downstream to the trailing edge. The assumption is then made that the boundary layer displacement area at the point where

$$\delta^* \cos \alpha = r_o \tag{37}$$

is defined by

$$\text{DAREA} = \pi \left\{ \left(r_{o_p} + \delta^* \cos \alpha_p \right)^2 - r_{o_p}^{\ 2} \right\} \tag{38}$$

where p refers to the point where equation (37) is first satisfied. This displacement area is then considered to remain constant from the point p to the trailing edge. The new "viscous" body coordinates in this region are then defined by

$$y_{new} = \left\{ \frac{\pi r_o^{\ 2} + \text{DAREA}}{\pi} \right\}^{\frac{1}{2}} \tag{39}$$

The second problem area occurs when the boundary layer separates from the body creating a separation bubble. This bubble must be accounted for in the creation of a "viscous" body if the flow about this configuration is to be predicted accurately. The simplest technique of modeling this separation bubble is to assume that the flow leaves the surface parallel to the free-stream direction, producing a cylindrical wake shape as shown in figure 8. This approach, however, gives decelerations in the flow at the junction of the body with the cylinder as shown in figure 9, which do not exist in the real flow field. To minimize this problem, a circular arc is used to fair the body into the separated cylinder.

This circular arc is defined by passing a circle through the last three "viscous" body coordinates defined prior to the separation point. The radius of this circle is then used to create a circular arc which is tangent to the "viscous" body at the point of separation. The center of this arc is then defined according to whether the surface slope of the body at separation is positive or negative.

If the surface slope is positive then the center is taken as the center of the circle passed through the three points as defined above. This center is defined by

$$x_c = x_{sep} + R \sin \left[\tan^{-1} \left|\frac{dy}{dx}\right|\right] \tag{40a}$$

$$y_c = y_{sep} - R \cos \left[\tan^{-1} \left|\frac{dy}{dx}\right|\right] \tag{40b}$$

where R = Radius of the circle

$\frac{dy}{dx}$ = Surface slope at separation

This arc is then used from the point of separation to either the end of the body or to the maximum point on the arc, where $dy/dx = 0$, as shown in figure 10a. If the maximum point of the arc occurs before the trailing edge of the body is reached then a cylinder is defined which extends from the maximum point of the arc to the trailing edge.

If the surface slope is negative then the circular arc is defined such that the center is located above the body. The center is then defined by

$$x_c = x_{sep} + R \left[\sin \tan^{-1} \left|\frac{\partial y}{\partial x}\right|\right] \tag{41a}$$

$$y_c = y_{sep} + R \left[\cos \tan^{-1} \left|\frac{\partial y}{\partial x}\right|\right] \tag{41b}$$

This arc is then used from the point of separation to either the end of the body or to the minimum point on the arc, where $dy/dx = 0$, as shown in figure 10b. If the minimum point of the arc occurs before the trailing edge of the body then a cylinder is defined which extends from the minimum point of the arc to the trailing edge of the body.

The above separated wake model has been derived from intuitive considerations rather than from first principals. It does, however, provide reasonable results, as will be shown in the subsequent discussion.

The base drag coefficient for blunt axisymmetric bodies is calculated using the method of Hoerner, reference 35. This approach is based on the assumption that the flow field behind a blunt base is basically a jet pump, in that, air flowing around the body leaves the trailing edge forming a cylindrical jet which attempts to pump away the stagnated air in the base region. However, since there is no air to replace this stagnated air, the pumping mechanism can only reduce the static pressure acting on the base. The effectiveness of this jet pump mechanism is controlled by the boundary layer thickness at the base since this region of lower momentum flow acts as a buffer between the stagnated air behind the base and the flow in the jet. Since the boundary layer thickness is directly related to the skin friction on the body, C_f, Hoerner used C_f to correlate with the base drag to develop an empirical approach to determine base drag. Figure 11 shows the correlation obtained by Hoerner for bodies whose base area is the same as the maximum area. This curve is represented by

$$C_{D_{BASE}} = .029/\sqrt{C_{f_{Forebody}}} \tag{42}$$

where the coefficients are based on the base area. Thus, once the skin friction on the forebody has been calculated in the boundary layer programs, then the base drag can be determined by equation 42.

This equation must be modified for boat-tailed bodies, that is, bodies whose base area is less than their maximum area. The mechanics of the base drag for these configurations do not change, but the calculation must take into account the reduced base area. This effect is taken into account by the following relationship:

$$C_{D_{BASE}} = C_{D_{BASE}} \cdot \left(\frac{d_{BASE}}{D_{MAX}}\right)^2 \qquad (42a)$$

and

$$\underset{\text{(BOAT TAIL)}}{C_{f_B}} = C_{f_B} \cdot \left(\frac{D_{MAX}}{d_{BASE}}\right)^2 \qquad (43b)$$

so

$$\underset{\text{BOAT TAIL}}{C_{D_{BASE}}} = \frac{.029}{\sqrt{C_{f_B}}} \cdot \left(\frac{d_{BASE}}{D_{MAX}}\right)^3 \qquad (43c)$$

A comparison of results calculated by the above method in ADAM with experimental force data from reference 35 is presented in figure 12. One of these cases is for a boat-tailed body and the other for a body whose base area is also the maximum area.

The experimental data used for this comparison as well as the configuration used for the analytical calculations are both subject to some discussion. The experimental base drag, taken from Figure 4 of Reference 35, originally came from an old German report which is not readily available. These base drag values were obtained from both force measurements and pressure measurements which unfortunately do not agree. Therefore, since it was felt that the force measurements were the more accurate, they were used in the comparison shown in Figure 12. In addition, no good definition of the configuration tested was available, therefore, the geometry used in the ADAM analysis was taken from the schematics shown in Reference 35. In light of these uncertainties the comparison presented in Figure 12 is fairly good in that even though the levels are different, the trends are the same. It should be noted that this comparison was used only because there is a singular lack of experimental data for blunt based axisymmetric bodies at low subsonic Mach numbers.

EXPERIMENTAL CORRELATIONS

Experimental results from three different configurations were selected to establish the extent of validity of the method presented in this report. These geometries consisted of a high fineness ratio body of revolution, and a sphere in both subcritical and supercritical flow regimes. These correlations, while limited to some extent by the scope of the present effort, do represent a wide range of axisymmetric flow conditions.

The body of revolution chosen was tested in the low speed wind tunnel at the Douglas Aircraft Company, (Reference 36), and is shown in figure 13. This model was composed of three sections; an elliptical nose section, a cylindrical control section, and a parabolic afterbody. The calculation done for this configuration used the wind tunnel flow properties, namely, U_∞ = 71.628 M/Sec (235 Ft/Sec), T_∞ = 288.3°K (519.0°R) and $R_L = 10.05 \times 10^6$. Boundary layer transition was fixed on the model and in the calculation at .03048 meters (1.2 inches) from the nose. This model was relatively large for the wind tunnel in which it was tested; wall effects, not accounted for in the original data reduction, were present. To correct for this, the model was run in the potential flow program in the presence of the wind tunnel walls as shown in figure 14. The effect of including the walls in the calculation is shown in the inviscid pressure distributions of figure 15. The final results for this configuration are shown in figure 16 where the calculated "viscous" results are compared to experimental data. The inviscid distribution is also shown for reference. In this particular case no separation occurred and so only one iteration, that is, two potential flow solutions and two boundary layer solutions, was necessary. The calculated "viscous" results agree very well with the experimental values except in the region of the nose. This discrepancy is not due to the calculation method, but rather is due to the model being too long for the wind tunnel test section resulting in the nose being in a different static pressure field than the rest of the body. The overall effect of viscosity on this configuration is seen to be small except in the region of the trailing edge. The body is so slender in this region that the boundary layer equations are no longer valid so the technique described in the calculation procedure was used to modify the viscous body. The results show a pressure osciliation in this modified region which is due to an unsmooth curvature distribution. However, the level of these pressures agree quite well with the experimental values.

The second case considered was that of a sphere in the supercritical flow regime, i.e., $R_D = 1 \times 10^6$. Since the boundary layer transition was forced to occur at an $X/D = .65$, there were regions of both laminar and turbulent flow present. The experimental data for this case were taken from references 37 and 38. The freestream velocity assumed for this case was 47.85 M/Sec (157 Ft/Sec). Figure 17 shows the sphere with the "viscous" body superimposed and figure 18 presents a comparison between the calculated "viscous" solution and experimental data. Note that while the calculated pressure distribution is in reasonably good agreement with the experimental values, the calculated separation point is .07 diameters further downstream than the experimentally measured value. The inviscid and "viscous" solutions for the local skin friction coefficient, C_f, are presented in figure 19. The "viscious" solution shown is the fourth iteration, i.e., the fifth potential flow solution, and appears to be the best solution possible for this configuration with the technique being used in the present method to simulate flow separation.

The last correlation to be presented is for the flow about a sphere in the subcritical regime, i.e., $R_D = 1 \times 10^5$, which is a purely laminar case. The experimental data is again taken from Reference 37. The freesteam velocity for this case was assumed to be 4.785 M/Sec (15.7 Ft/Sec). The calculated "viscous" body is shown in figure 20 while a comparison of the "viscous" pressure distribution to experimental data is shown in figure 21. The calculated "viscous" pressures are in close agreement with the experimental values with some slight over-prediction in the separated region. The calculated separation point is only .03 diameters further downstream than the experimental value which is excellent considering the large effect that viscosity has on this configuration. Figure 22 presents the inviscid and "viscous" solutions for the local skin friction coefficient for this case.

CONCLUDING REMARKS

A method has been presented for the computation of the viscous flow field about axisymmetric bodies at zero angle of attack in incompressible flow. This computing program requires only the specified body geometry and desired flow conditions as input. The appropriate theory has been discussed and correlations between theoretical and experimental results presented.

The flow field about axisymmetric bodies at zero angle of attack with no flow separation is well defined and can be computed accurately by the present method. When flow separation occurs, the flow field is no longer amenable to analytical treatment. Currently, methods do not exist to calculate the flow field within a separated region; it is therefore necessary to resort to empirical methods to account for flow separation. Since there is almost a complete lack of experimental data concerning the behavior of separated regions, any empirical methods must necessarily be somewhat crude. The most sophisticated model for separation currently available is due to Jacob (References 39 and 40) and is strictly for two-dimensional airfoils. An unsuccessful attempt was made in Reference 41 to adapt Jacob's approach to axisymmetric configurations. The conclusions of Reference 41 indicated that the assumed boundary conditions needed to be modified if this approach was to be used for axisymmetric flow. It is proposed that the Douglas-Neumann program be used to pursue this approach at modeling separation. This potential flow program is ideal for attempting to use Jacob's technique since it already has the ability to specify a non-uniform flow distribution over all or part of a configuration; therefore, only suitable boundary conditions would have to be added to the program. It is felt that this approach can be successful in modeling separation if care is taken in developing the distribution of non-uniform velocity as well as specifying the proper boundary considerations.

The further extension of this model to the calculation of flow about axisymmetric bodies at angle of attack is also possible. The potential flow routine contained in the present method has the capability of predicting the flow field about non-lifting bodies at angle of attack by combining the streamflow and the crossflow solutions. The boundary layer analysis would require the replacement of the routine in the present method by a three-dimensional technique, which is currently not available. However, it is felt that a good approximation to the boundary layer calculations can be made by the small crossflow program of Reference 42.

One area of primary concern in extending the method to include an angle of attack capability is the determination of the separation line about the body. The present method of predicting separation for two-dimensional bodies and for axisymmetric bodies at zero angle of attack is to find the location where the skin friction goes to zero. It has been shown in several studies, including those reported in References 43 and 44, that this condition does not apply in three-dimensional flows because the skin friction along a separation line is not necessarily zero. Therefore, some method of determining the separation line for axisymmetric bodies at angle of attack must be developed. It is proposed that the present method could be extended to calculate the "viscous" flow about axisymmetric bodies at angle of attack when no flow separation is present. This method could then be used to assist in the development of a procedure for determining the separation line location. Once the location of the separation line is known then a model could be developed for analyzing the viscous flow about the separated body. The development of such procedures is not a simple task and considerable effort would have to be expended; but the reward for accomplishing this task is an advance in the ability to calculate the real flow about arbitrary three-dimensional bodies which is our ultimate goal.

APPENDIX A

INPUT INFORMATION FOR ADAM COMPUTER PROGRAM

This part of the report contains the necessary information to input data to the ADAM computer program. The input data is broken into three sections: smoothing, potential flow, and viscous flow. These sections can be used together in the iterative fashion described in the main text, or the potential flow and viscous flow sections may be used independently. A detailed card-by-card description of all input quantities is given followed by a set of input forms which can be used to facilitate the loading of the input data into the program.

Input Instructions

The Adam program requires one system control card followed by the required sets of data cards for each program option to be executed. The sets of data furnished must be in the same order as the options are specified on the system control card. If an iteration is desired the system control card is repeated along with the necessary other data cards.

The general scheme used in describing the input data is shown below:

Column	Code	Routine Format	Explanation

Column - Column indicates the starting position on the card for each data field.

Code - The "code" gives the FORTRAN name used in the read statement by the program.

Routine - "Routine" indicates the subroutine where the data is read.

Format - The parameter "FORMAT" which is given right under the routine name, indicates the FORTRAN format of the data read statement field. The parameter I5 would indicate that the parameter is an integer in a field that is 5 columns wide. Integers should be punched on the right side of the field (right justified). The parameter F10.0 would indicate a fixed point number punched with a decimal point (i.e., -12.354). The number may be punched anywhere in the field indicated irrespective of the decimal point location indicatied by the format. The parameter E12.6 would indicate a floating point number punched with a decimal point (i.e., 5.0 x 10^6). The number must be punched to the right of the field in the manner 5.0E+06.

Explanation - The description of the input data is given under "explanation".

SUSTEM CONTROL DATA CARD (This card must be the first card in the data deck)

Column	Code	Routine Format	Explanation
4	IGEOM	MAIN I1	Smoothing option flag =0 no smoothing is desired =1 smoothing is desired
8	INEUM	MAIN I1	Potential flow option flag =0 No potential flow solution is desired =1 Potential flow solution is desired
12	IBOUND	MAIN I1	Boundary layer option flag =0 No boundary layer solution is desired =1 Boundary layer solution is desired
16	ITER	MAIN I1	"Viscous" body formation flag =0 No "viscous" body is formed =1 "Viscous" body is formed
17-20	IFINSH	MAIN I4	Termination Flag =0 Another case expected =9999 Program will stop after exercising all options specified above

SMOOTHING SECTION

These cards required if IGEOM = 1 on system control card. This section is used to smooth body geometry data before it is input to the potential flow program.

Smoothing Control Card

Column	Code	Routine Format	Explanation
2	NPTS	SMOOTH I3	Number of input data points for this configuration. NPTS must be ≤ 100
8	ITAPE	SMOOTH I1	Data source flag =0 Data input on unit 5 (card input) ≠0 Data input on unit 1. This is used for a case where a "viscous" body generated by the iteration procedure is being read.

Geometry Data Input Cards

These cards are input only if ITAPE = 0.

Column	Code	Routine Format	Explanation
x-coordinate cards			

1-10
11-20
21-30
etc.
	x(1)
	x(2)
	x(3)
		SMOOTH 6F10.0	x-coordinates starting at the leading edge and proceeding along the upper surface to the trailing edge. Input 6 x-values on each card. The numbers of x-values must be equal to NPTS.

y-coordinate cards			

1-10
11-20
21-30
etc.
	y(1)
	y(2)
	y(3)
		SMOOTH 6F10.0	y-coordinates to correspond to the above x-locations. y-values must be positive. Input 6 values per card.

POTENTIAL FLOW SECTION

These cards required if INEUM = 1 on system control card. The input geometry for this program may be obtained from the geometry storage unit (10) as generated by the smoothing section, or it may be input directly on unit 5. Thus, this program may be operated as a separate entry if so desired. The program saves the geometry data element midpoints with the corresponding pressure coefficients on unit 3 for input to the boundary layer routine and it saves the basic non-dimentional input Neumann coordinates on unit 1 for use if a "visous' body is desired.

Title Card

Column	Code	Routine Format	Explanation
1	HEDR	PART1 10A6	Title of case. May be any characters input in the first 60 columns of card;
63	CASE	PART1 I6	Case number
77	PSF	PART1 I6	Additional identifier for this case.

Flag Card

Card columns 1-30 when punched with any non-zero integer, activate flags that indicate the following:

Column	Code	Routine Format	Explanation
1	NB	PART1 I1	The number of bodies input. Normally set equal to 1. $1 \leq NB \leq 5$
2	NNU	PART1 I1	The number of non-uniform onset flows. Normally set equal to 0.
3	FLG03	PART1 I1	Axisymmetric flow flag. =0 No axisymmetric stream-flow solution calculated. =1 Axisymmetric streamflow solution is calculated. Normally set equal to 1

Flag Card (Continued)

Column	Code	Routine Format	Explanation
4	FLG04	PART1 I1	Cross flow flag. =0 No cross flow solution is calculated =1 Cross flow solution is calculated Normally set equal to 0
5	FLG05	PART1 I1	Off-body point flag =0 No off body points input =1 Off body points are input This flag allows the velocity at points off the body surface to be determined.
6	FLG06	PART1 I1	Basic data formation flag =0 A full case will be done =1 The basic data, i.e., midpoints, normals, etc. will be formed and printed. No velocities will be calculated.
7	FLG07	PART1 I1	Ellipse generator option =0 Body coordinates will be input =1 An ellipse is generated using data input later. No body coordinates are input
8	FLG08	PART1 I1	Matrix print flag =0 Coefficient matrices are not printed. =1 Coefficient matrices will be printed. Normally set equal to 0
11	FLG11	PART1 I1	Perturbation velocity flag =0 Normal case =1 No onset flow used. Only perturbation velocities are calculated.
12	FLG12	PART1 I1	Potential matrix solution * =0 Normal case =1 A potential matrix is solved

Flag Card (Continued)

Column	Code	Routine Format	Explanation
13	FLG13	PART1 I1	Matrix solution flag =0 No matrix solution done =1 Matrix solution performed Normally set equal to 1.
14	FLG14	PART1 I1	Prescribed tangential velocity flag * =0 Normal case =1 Tangential velocities are specified
15	FLG15	PART1 I1	Strip ring vorticity flag * =0 Normal case =1 A vorticity distribution is formulated.
16	FLG16	PART1 I1	Axisymmetric uniform flow flag =0 Normal case =1 Axisymmetric uniform flow solution is omitted Normally set equal to 0.
17	FLG17	PART1 I1	Crossflow uniform flow flag =0 Normal case =1 Crossflow uniform flow solution is omitted. Since FLG04 is normally = 0 then so is FLG17 normally set equal to 0.
18	FLG18	PART1 I1	Surface vorticity flag * =0 Normal case =1 Surface vorticity is generated.
19	FLG19	PART1 I1	Prescribed vorticity Flag * =0 Normal case =1 A prescribed vorticity is input
20	FLG20	PART1 I1	Total vorticity flag * =0 Normal case =1 Total vorticity calculated

Flag Card (Continued)

Column	Code	Routine Format	Explanation
21	FLG21	PART1 I1	Extra crossflow flag * =0 Normal case =1 Extra crossflow option used
22	FLG22	PART1 I1	Generated boundary condition flag * =0 Normal case =1 Boundary conditions generated
23	FLG23	PART1 I1	Ring wing option flag * =0 Normal case =1 Ring wing option used
28-29	NIN	PART1 I2	Tape input flag =0, 10 Data input on unit 10 from smoothing program =5 Data input from unit 5 (card input)
30	ITER	PART1 I1	Iteration tape flag =0 x/c, y/c transformed data saved on unit 15 =1 x/c, y/c transformed data not saved. This flag is necessary because for a "viscous" body to be formed, the coordinates of the original unmodified body must be saved. Therefore, for the first case set ITER = 0. For subsequent iterations we do not want to use the modified bodies to form new bodies so set ITER = 1.

* These flags are for special options which are discussed in the main text. They are never used for a normal axisymmetric calculation. Therefore, set them equal to zero.

Chord Card

Column	Code	Routine Format	Explanation
1	CHORD	PART1 F10.0	Reference chord length used to non-dimensionalize x and y coordinates
11	MN	PART1 F10.0	Mach number (MN < 1.0) use to approximate effect of compressibility (Gothert's rule)
21	TCNST	PART1 F10.0	This is a constant which is used for the value of the tangential velocity if this option is desired.

Body Transformation Card

Column	Code	Routine Format	Explanation
8	NN	BASIC1 I3	The number of input points on this body. NN ≤ 100
11	MX	BASIC1 F10.0	A factor used to multiply all x-coordinates. MX is assumed equal to 1 if no value is input.
21	MY	BASIC1 F10.0	A factor used to multiply all y-coordinates. MY is assumed equal to 1 if no value is input.
31	THETA	BASIC1 F10.0	An angle (in degrees) through which all points of a body are to be rotated about the origin in the clockwise direction.
41	ADDX	BASIC1 F10.0	A constant to be added to all x-coordinates
51	ADDY	BASIC1 F10.0	A constant to be added to all y-coordinates

Body Control Card

Column	Code	Routine Format	Explanation
10	BDN	BASIC1 I1	Body sequence number. This program will handle up to 5 bodies.
20	SUBKS	BASIC1 I1	Subcase Flag. =0 Normal case =1 Use unmodified coordinates of the previous case.
30	NLF	BASIC1 I1	Non-lifting flag =0 Body is non-lifting (normal case) =1 Body is lifting (this is used in special option)
31	XE	BASIC1 F10.0	Value of major semi-axis for use by ellipse generation option.
41	YE	BASIC1 F10.0	Value of minor semi-axis for use by ellipse generation option Note: if XE = YE a sphere will be formed.

Geometry Data Cards

The body geometry data cards are included only if the input parameters NIN = 5 and FLG07 = 0 on the flag card. If NIN = 0 or 10 then the data is read from unit 10. If NIN = 5 and BDN = 0, then the following cards contain the x-y coordinates of off-body points instead of x-y geometry data. The number of either geometry data point or off-body points must be equal to NN.

x-Coordinate cards (six values per card)

Column	Code	Routine Format	Explanation
1	TX1(1)	BASIC1 6F10.0	x-coordinates of body input from leading to trailing edge.
11	TX1(2)		
21	TX1(3)		

y-Coordinate cards (six values per card)

Column	Code	Routine Format	Explanation
1	TY1(1)	BASIC1 6F10.0	y-Coordinates of body which correspond to the x-values above. y values must be positive.
11	TY1(2)		
21	TY1(3)		
etc.			

NOTE: Each body input, including the off body points, requires the body transformation card, the body control card, and may also require the geometry data cards depending on the input flags. This is the stopping place for a normal axisymmetric case. The following cards are input only if one of the special options is required.

Tangential Velocity Data (six values per card)

These cards are input only if FLG14 ≠ 0 and TCNST = 0.0

Column	Code	Routine Format	Explanation
1	TG(1)	BASIC1 6F10.0	Specified tangential velocities at element midpoints.
11	TG(2)		
21	TG(3)		
etc.			

Non-uniform Flow Cards (six values per card)

These cards are input only if NNU ≠ 0.

Column	Code	Routine Format	Explanation
6	NUM	BASIC2 I5	Non-uniform flow identification number.

Non-uniform Flow Cards (Continued)

Column	Code	Routine Format	Explanation
19	MSF	BASIC2 I2	If MSF = 0 the flow velocities N_o, T_o will be used for the axisymmetric case only. If MSF = 1 the flow velocities N_o, T_o will be used for the cross flow case only. If MSF > 1 the flow velocities will be used for both axisymmetric and cross flow cases.
21	TYPE	BASIC2 F10.0	Flag which specifies the type of input flow velocities at each mid-point. If TYPE > 0.0, the velocities are input as x & y components. If TYPE = 0.0 the velocities are input as normal & tangential components. If TYPE < 0.0 the automatic generation of the flow due to a rotating body is used.
31	FG	BASIC2 F10.0	Constant used by the flow generator. Type must be less than 0.0.

The following cards are input only if NNU $\neq$ 0 and TYPE $\neq -1.0$.

Normal velocity cards (six values per card)

Column	Code	Routine Format	Explanation
1	NO(1)	BASIC2	This is either the x or normal velocity component depending on the value of type above. These values must be in sequence with the coordinate data. If the x component is input it is defined as positive to the right. If the normal velocity is input it is positive if it is to the interior of the body. NN-1 values are input.

Tangential Velocity Cards (six values per card)

Column	Code	Routine Format	Explanation
1	TO(1)	BASIC2 6F10.0	This is either the y or tangential velocity component depending on the value of type above. These values must correspon to the NO values above. If the y component is input it is defined as positive if it is orientated upwards. If the tangential velocity is input it is positive if the flow field is to the left of the vector representing the tangential velocity.

VISCOUS FLOW SECTION

These cards required if IBOUND = 1.

BOUNDARY LAYER PROGRAM

The geometry and pressure distribution data required by this program may be input directly on cards (Unit 5), or read from the data save unit (Unit 3) as generated by the Neumann program.

HEADER CARD

This card is supplied purely for description purposes.

Column	Code	Routine Format	Explanation
1-60	TITLE	INPT 15A4	Description of input
61	CASE	INPT A4	Case number

Flag Control Card

This card contains flags which control the type of flow to be considered and the form of the input.

Column	Code	Routine Format	Explanation
1	NXT	INPT I4	The number of the x-station where the flow goes turbulent measured from the stagnation point (i.e., the leading edge for axisymmetric bodies at zero angle of attack) if transition is to be calculated by the program set NXT to be one greater than the number of points input.
5	LG16	INPT I1	Transition flag =0 Boundary layer transition point is input =1 Boundary layer transition point is computed. Set NXT to be greater than number of points input.

Flag Control Card (Continued)

Column	Code	Routine Format	Explanation
6	LG17	INPT I1	Transition control flag =0 Transition is instantaneous =1 Transition is gradual (transitional region used)
7	LG18	INPT I1	Transverse curvature flag =0 No transverse curvature correction used. =1 Transverse curvature corrections applied.
8	LG32	INPT I1	Print control flag =0 Print using long format (with velocity profiles) =1 Use short printout (no velocity profiles)
9	LG26	INPT I1	Velocity input control flag =2 Velocity ratio (U_e/U_∞) is input =3 Pressure coefficient (c_p) is input.
10	LG40	INPT I1	Unit input flag for geometry and velocity data. =0 Data read from unit 3 as generated by the potential flow program. ≠0 Data read from cards (unit 5)
11	LG41	INPT I1	System of units FLAG =0 English system of units =1 Internation system of units

Flow Condition Card

Column	Code	Routine Format	Explanation
1	TI	INPT F10.0	Reference static temperature used to compute the reference fluid properties. If TI is input as zero then TI is set equal to either 288.33°K or 519°R depending on FLAG LG41

Flow Condition Card (Continued)

Column	Code	Routine Format	Explanation
11	RMI	INPT F10.0	Reference or free-stream Mach number. =0.0 UI is input next ≠0.0 UI is computed from RMI.
21	UI	INPT F10.0	Reference or free-stream velocity =0.0 M_∞ is input above ≠0.0 M_∞ input as zero above.
31	FK	INPT F10.0	Flow index =0.0 2-D flow assumed =1.0 Axisymmetric flow assumed
41	RL	INPT F10.0	Chord or reference length
51	RI	INPT E12.0	Reynolds number/foot $R_c/\ell = \frac{U_\infty}{\nu}$ If CHORD = 1.0 then RI must be Reynolds number based on CHORD. NOTE: The input of either Mach number or freestream velocity is for convenience only. This program is entirely incompressible.

Radius card

Column	Code	Routine Format	Explanation
1	ROMAX	INPT F10.0	Maximum radius of body. This is used to obtain frontal area for skin friction calculation.
11	DETA1	INPT F10.0	Initial step size of boundary layer velocity profile grid. For a case which contains turbulent flow set DETA1 = .005.

Radius card (continued)

Column	Code	Routine Format	Explanation
21	VGP	INPUT F10.0	VGP is the growth factor for the boundary layer velocity profile grid; for cases with turbulent flow set equal to 1.14. NOTE: For laminar cases the boundary layer velocity profile grid may be made constant if VGP = 1.0 is input. However, if this is done the minimum value of DETA1 that can be input is approximately .10. This can be calculated if the value of the transformed boundary layer thickness, ETAINF, in known. Then DETA1 becomes $\text{DETA1} = \frac{\text{ETAINF}}{100}$

Geometry-Pressure Distribution Cards

These cards input only if LG40 ≠ 0.

Point Number Card

Column	Code	Routine Format	Explanation
1	NXM	INPT I4	Number of data points to be input. Maximum of 100 points allowed.

x-Coordinate Data Cards

Column	Code	Routine Format	Explanation
1	XS(1)	INPT 6F10.0	x-coordinate points input from leading to trailing edge input 6 points per card. Number of points = NXM
11	XS(2)		
21	XS(3)		
etc.			

y-Coordinate Data Cards

Column	Code	Routine Format	Explanation
1	YS(1)	INPT 6F10.0	y-coordinate points corresponding to x-coordinates above input 6 points per card.

y-Coordinate Data Cards (continued)

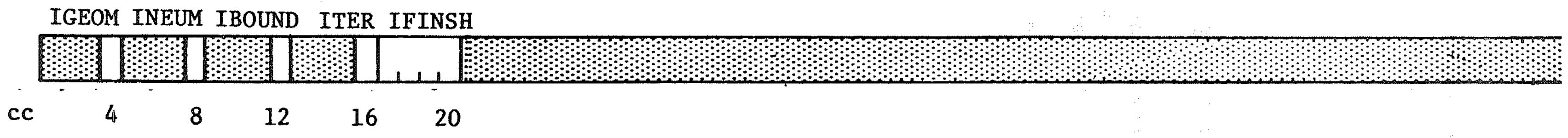
Column	Code	Routine Format	Explanation
11	YS(2)		
21	YS(3)		
etc.			

Pressure Distribution Cards

Column	Code	Routine Format	Explanation
1	UE(1)	INPT 6F10.0	Velocity-pressure-distribution points corresponding to x-points input above input 6 points per card. If LG26 = 2 u_e/U_∞ input; LG26 = 3 c_p input
11	UE(2)		
21	UE(3)		
etc.			

A D A M

AXISYMMETRIC DESIGN AND ANALYSIS METHOD



SYSTEM CONTROL DATA CARD

cc 4	IGEOM	=	0	NO SMOOTHING REQUIRED	=	1	FIVE-POINT SMOOTHING USED
cc 8	INEUM	=	0	NO POTENTIAL FLOW	=	1	NEUMANN ROUTINE USED
cc 12	IBOUND	=	0	NO BOUNDARY LAYER SOLUTION DESIRED	=	1	BOUNDARY LAYER SOLUTION WILL BE DONE
cc 16	ITER	=	0	NO "VISCOUS" SOLUTION IS DESIRED	=	1	A "VISCOUS" BODY WILL BE CREATED
cc 17	IFINSH	=	0000	ANOTHER CASE IS EXPECTED	=	9999	THIS IS THE LAST CASE

Instructions to Keypunch:
Do not punch blank columns

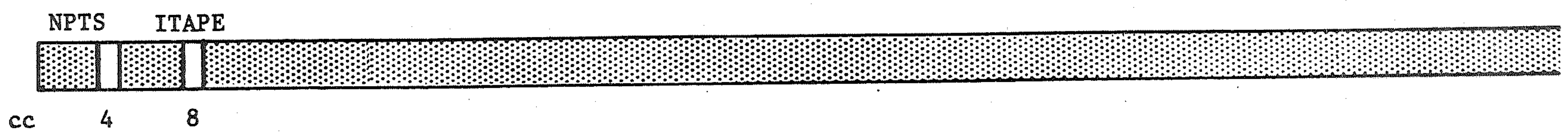
ENGINEER________________ PHONE________

DATE____________________ PAGE___OF____

A D A M

SMOOTHING PROGRAM

SMOOTHING CONTROL CARD



Instructions to Keypunch
Do not punch blank columns

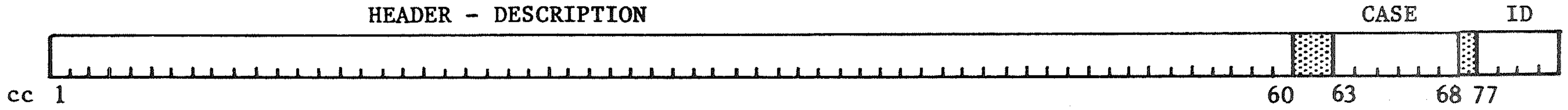
ENGINEER____________________ PHONE__________

DATE________________________ PAGE____ OF____

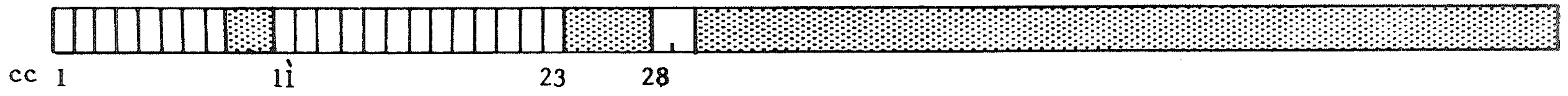
ADAM

NEUMANN POTENTIAL FLOW PROGRAM

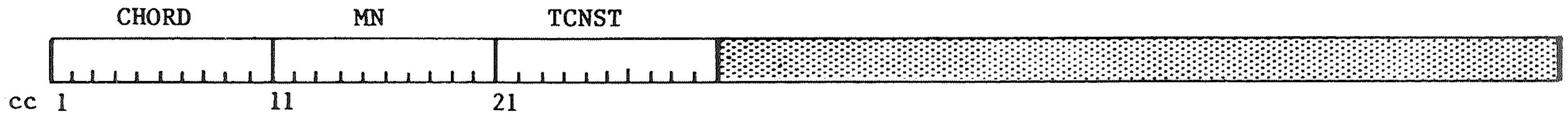
HEADER CARD



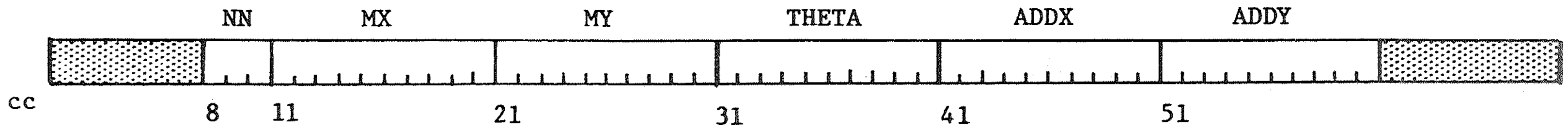
FLAG CARD



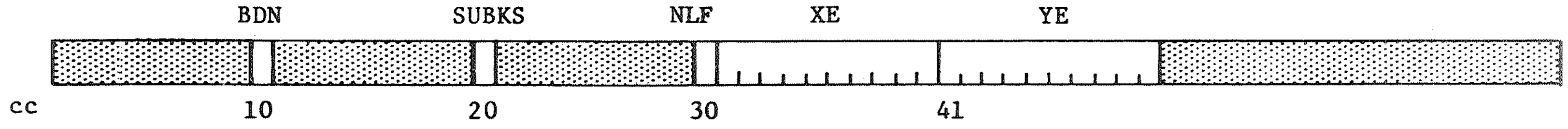
CHORD CARD



BODY TRANSFORMATION CARD



BODY CONTROL CARD



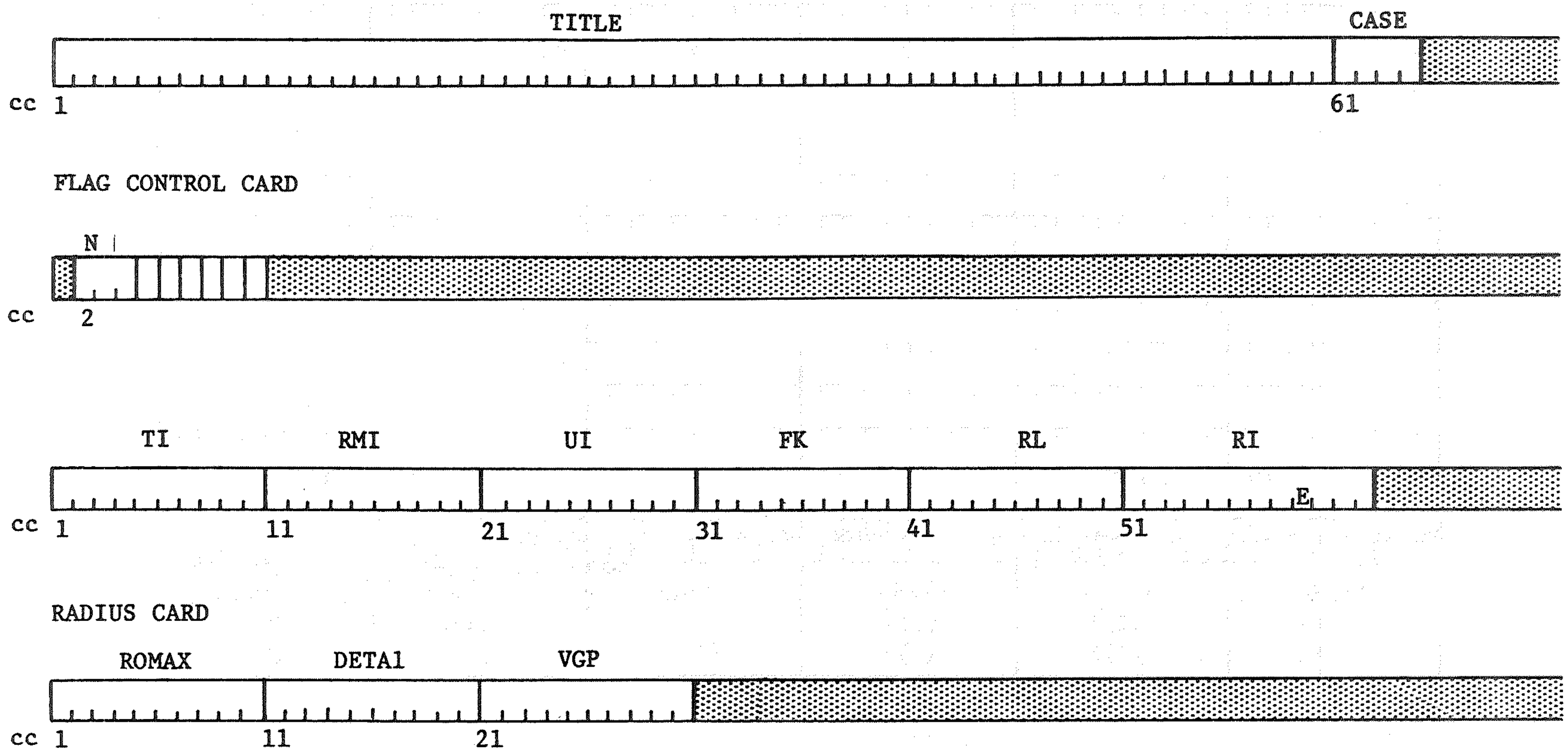
ENGINEER ____________________ PHONE ________

DATE ____________________ PAGE ____ OF ____

Instructions to Keypunch
Do not punch blank columns

A D A M

BOUNDARY LAYER PROGRAM



ENGINEER__________ PHONE__________

DATE__________ PAGE_____ OF_____

Instructions to Keypunch
Do not punch blank columns

Instructions to Keypunch
Do not punch blank columns

ENGINEER ______________ PHONE ______________

DATE ______________ PAGE _____ OF _____

COORDINATE DATA OR VELOCITY CARDS

	X(I) Y(I) CP(I)	X(I) Y(I) CP(I)	X(I) Y(I) CP(I)	X(I) Y(I) CP(I)	X(I) Y(I) CP(I)	X(I) Y(I) CP(I)
cc	1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60

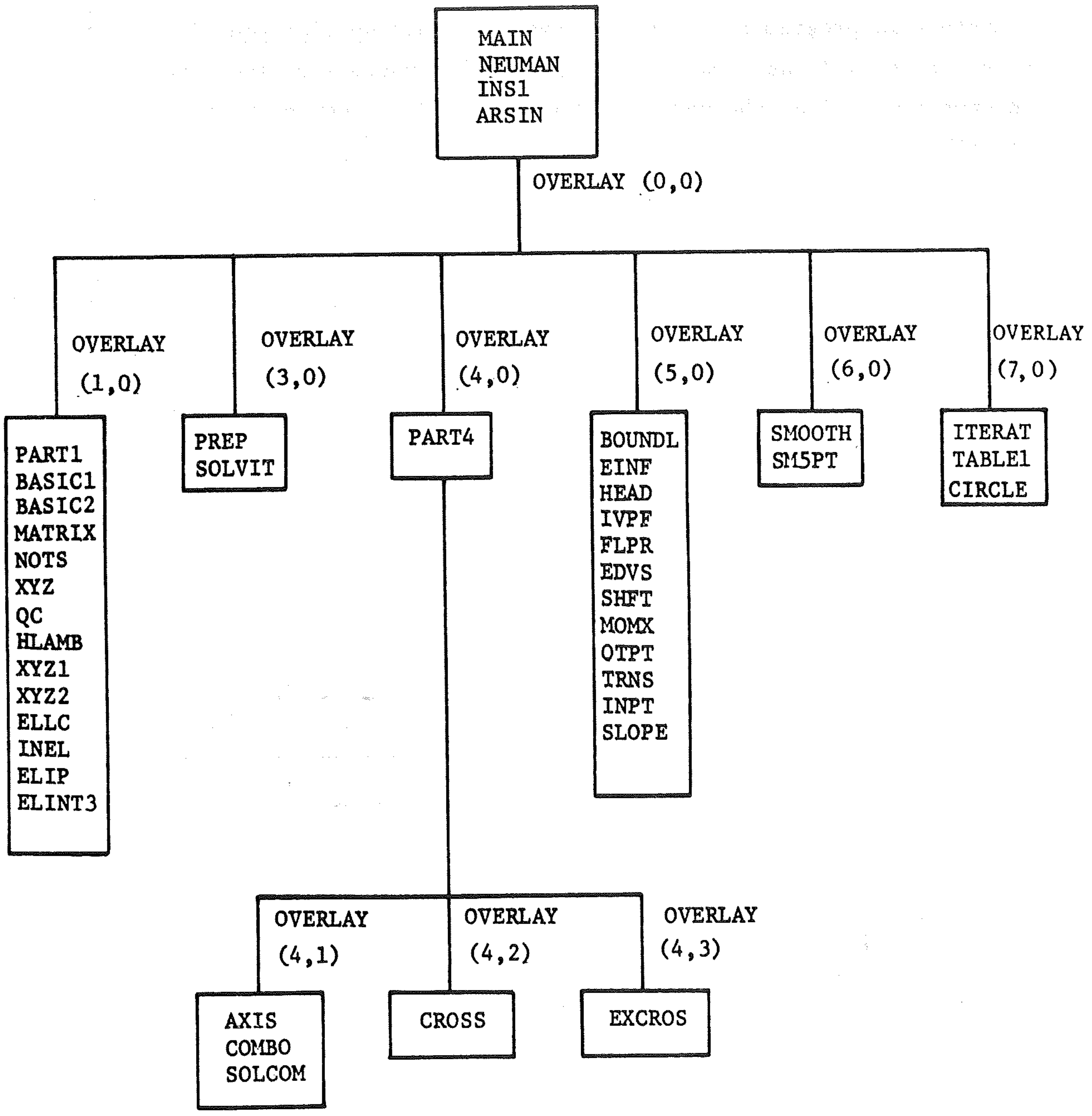
THIS INPUT FORM IS THE SAME FOR GEOMETRY DATA INPUT TO THE SMOOTHING ROUTINE, THE POTENTIAL FLOW PROGRAM, AND THE BOUNDARY LAYER ROUTINE. THIS FORM IS ALSO USED FOR VELOCITY DATA INPUT TO THE BOUNDARY LAYER PROGRAM AND FOR THE NON-UNIFORM VELOCITY DATA WHICH CAN BE INPUT TO THE POTENTIAL FLOW ROUTINE.

APPENDIX B

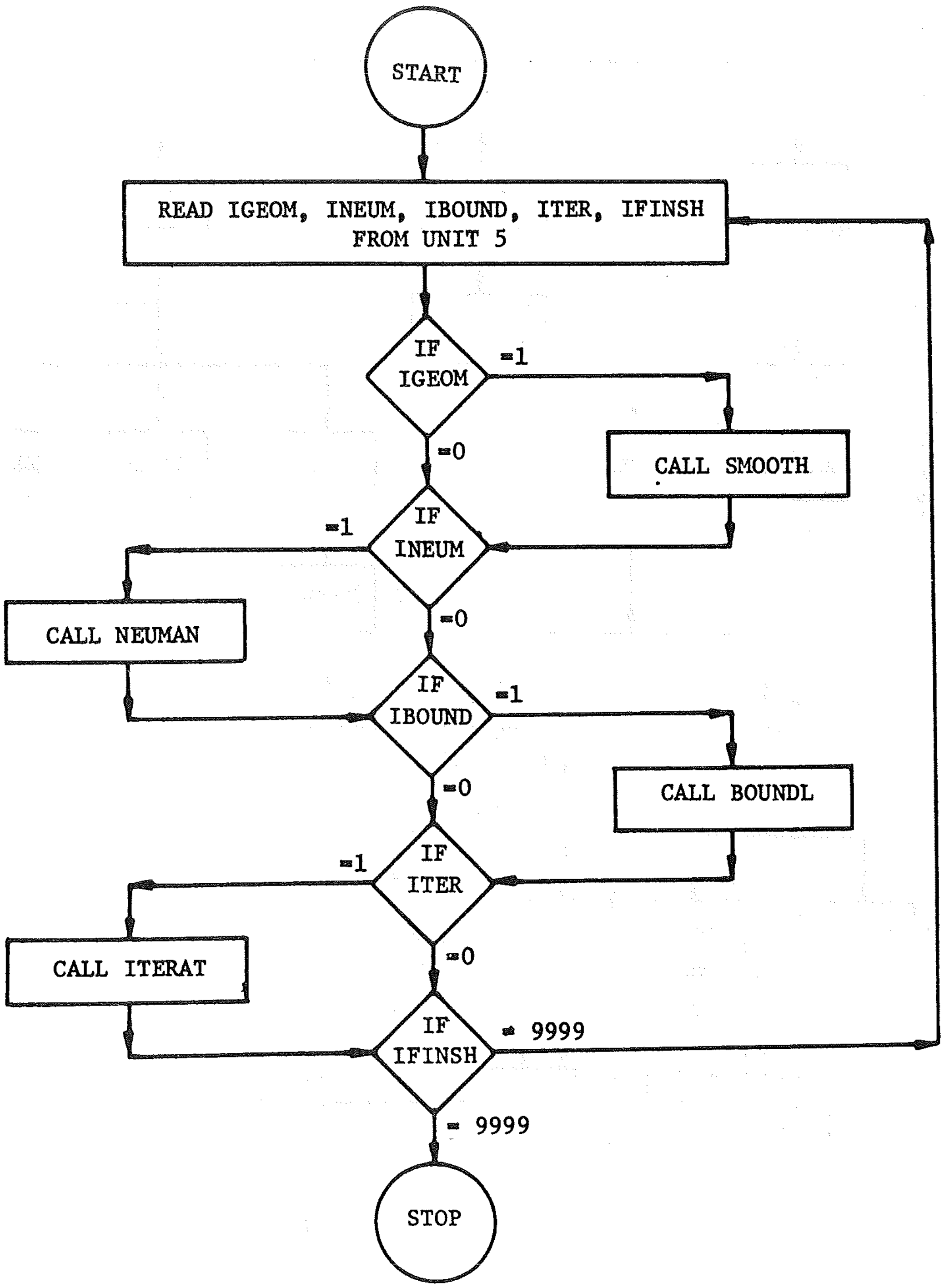
CONTROL INFORMATION FOR ADAM COMPUTER PROGRAM

This part of the report contains the necessary control information to operate this program on a computer system. This section contains the overlay structure as well as flow charts of the main subroutines including input flow information. Also, the various data sets used between main programs are described.

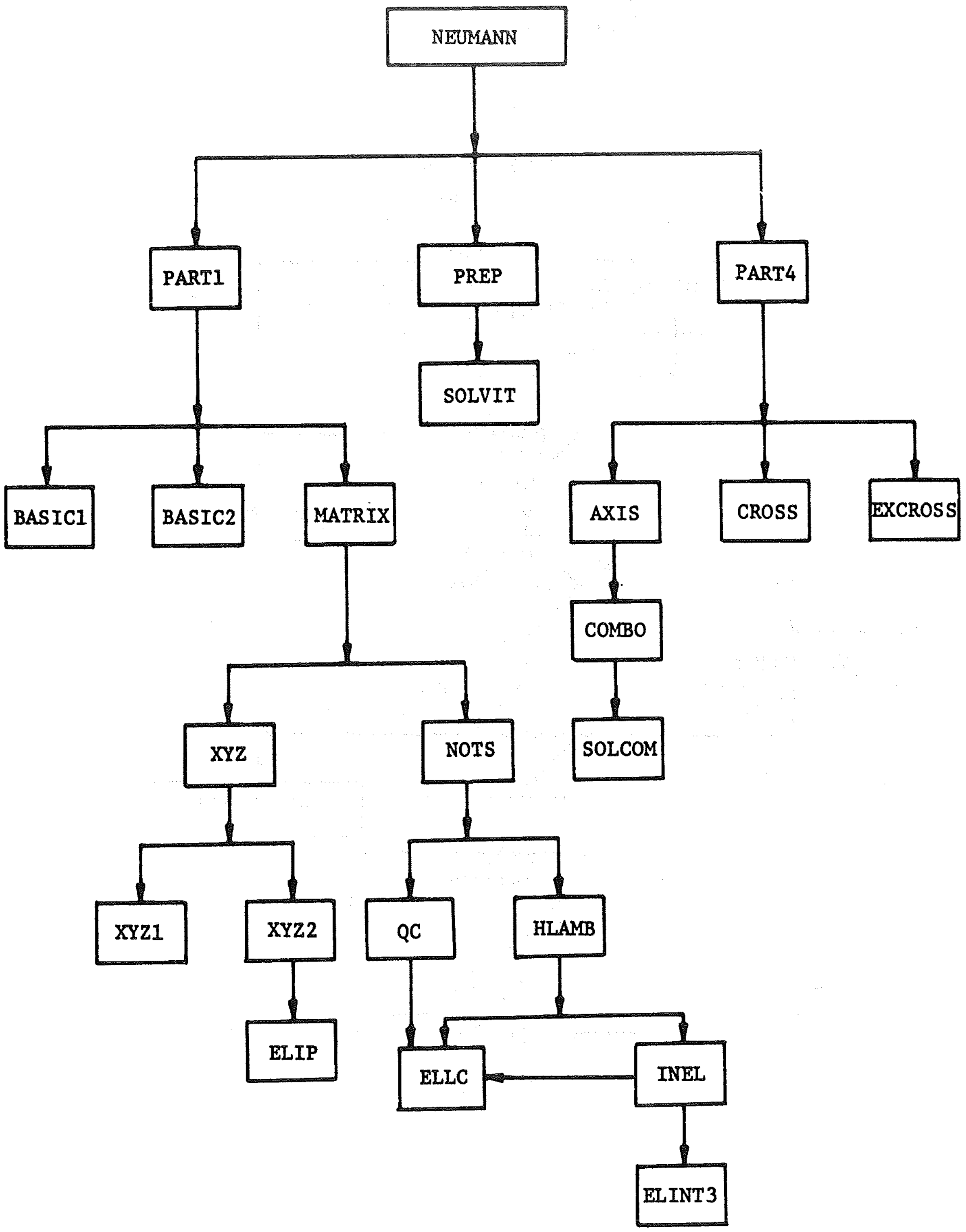
OVERLAY STRUCTURE OF ADAM



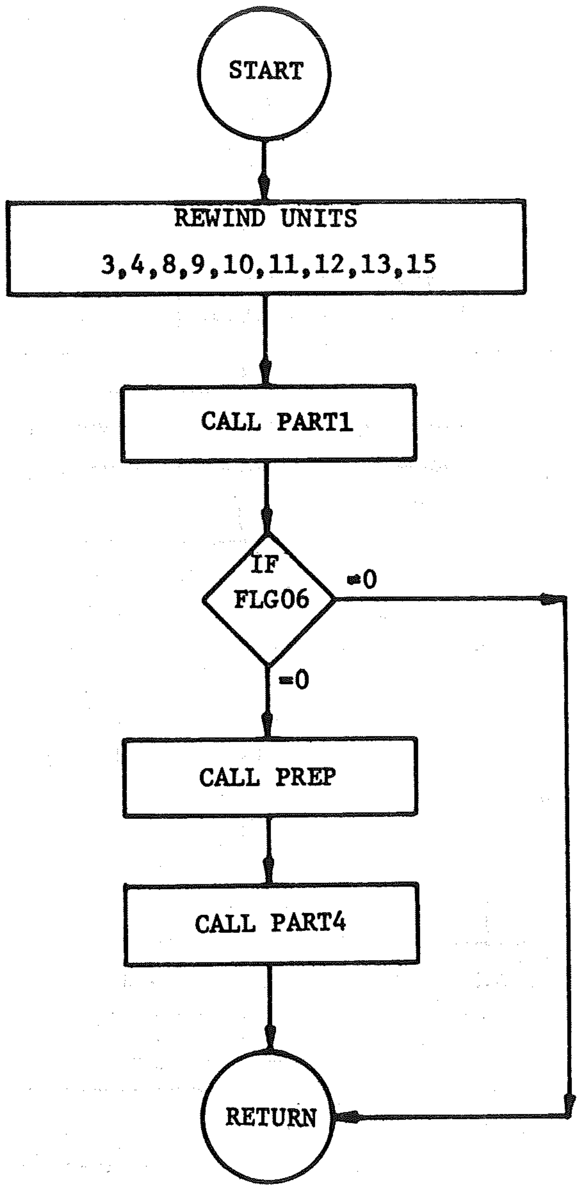
MAIN PROGRAM



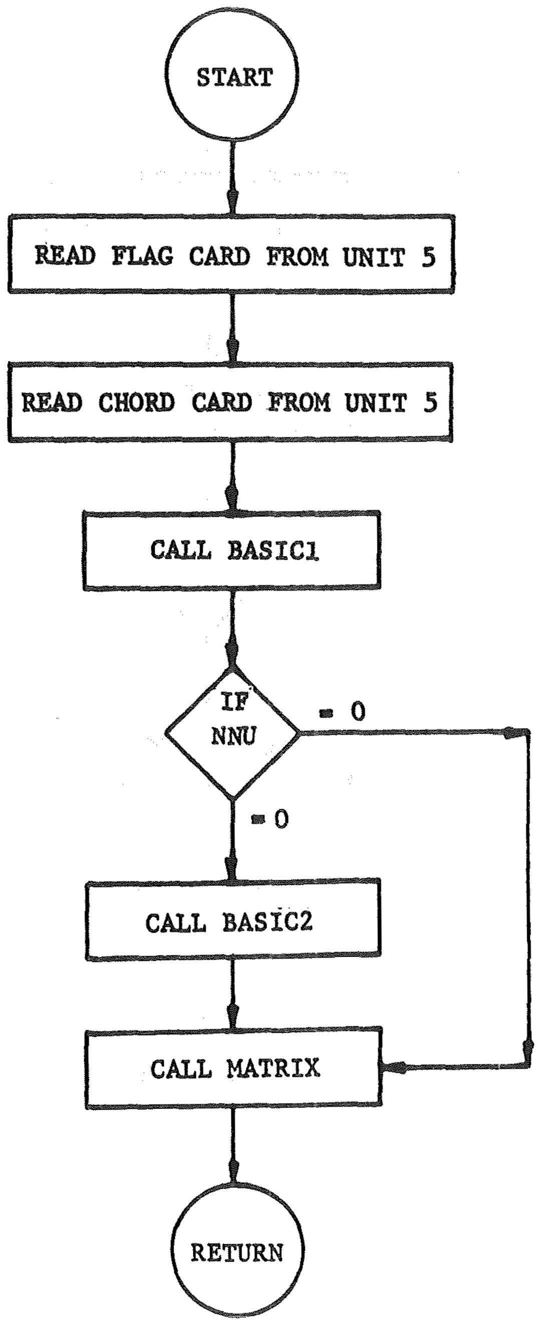
FUNCTIONAL ORGANIZATION OF NEUMANN PROGRAM



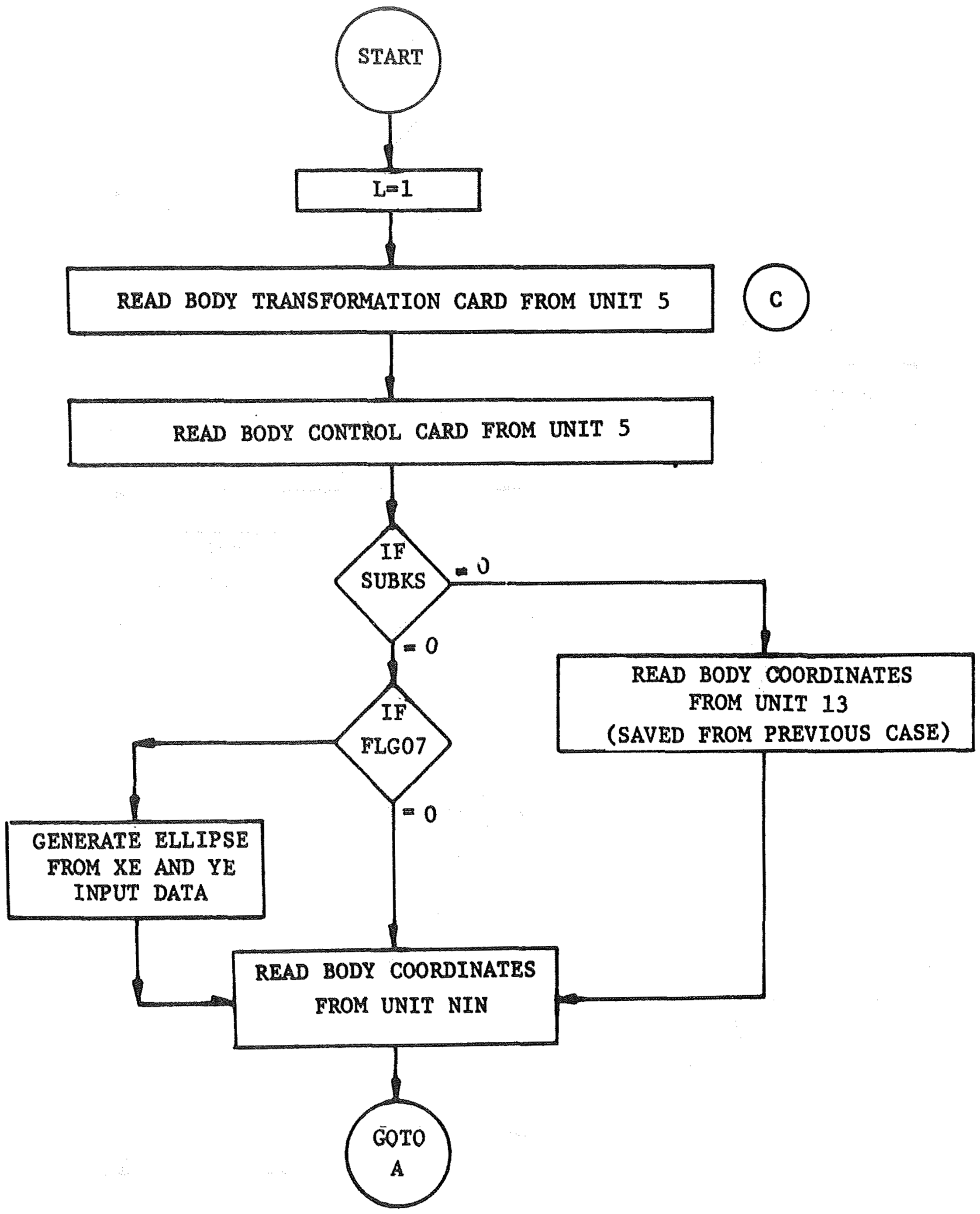
BASIC FLOW CHART FOR SUBROUTINE NEUMANN



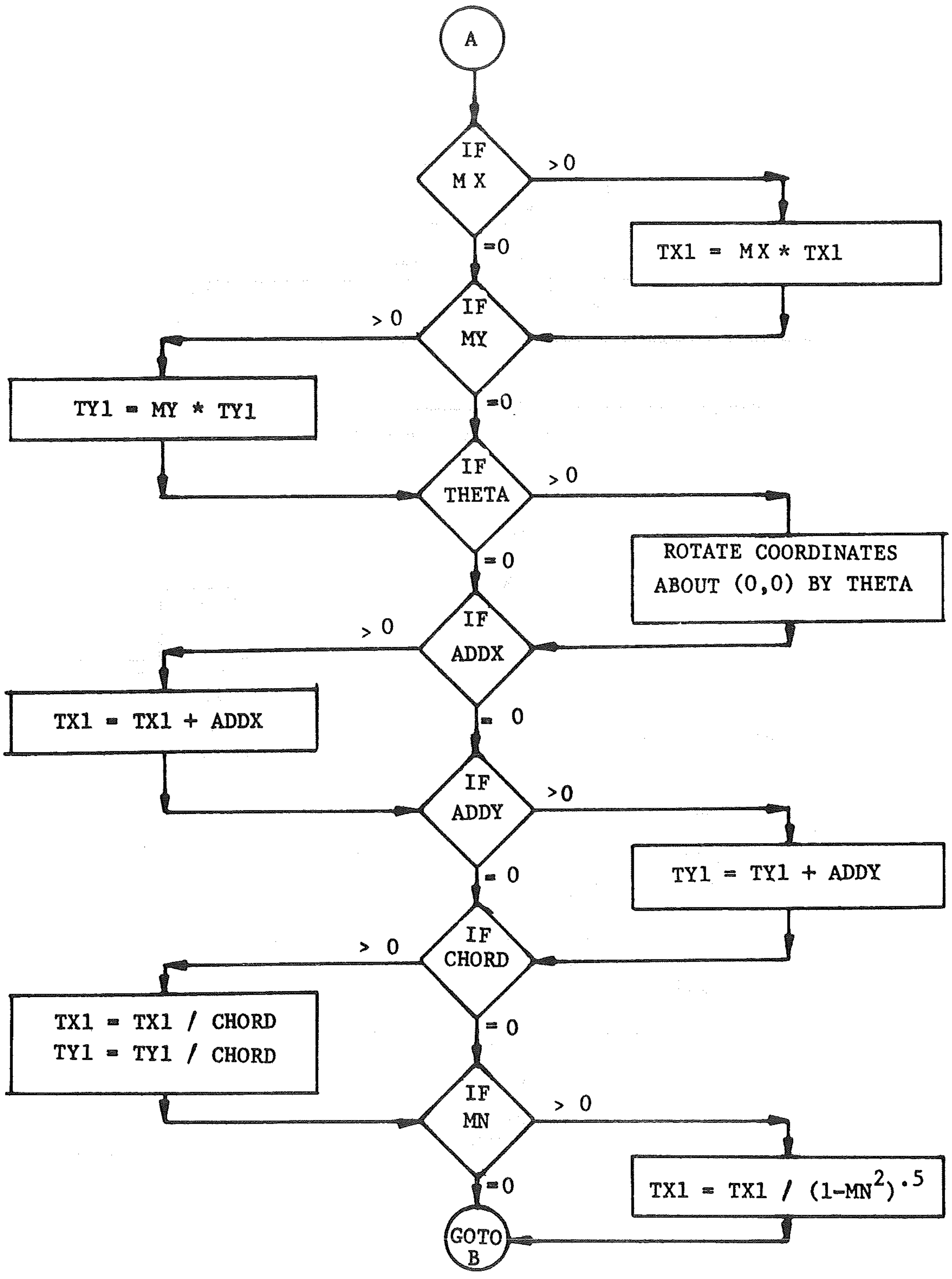
BASIC FLOW CHART FOR SUBROUTINE PART1

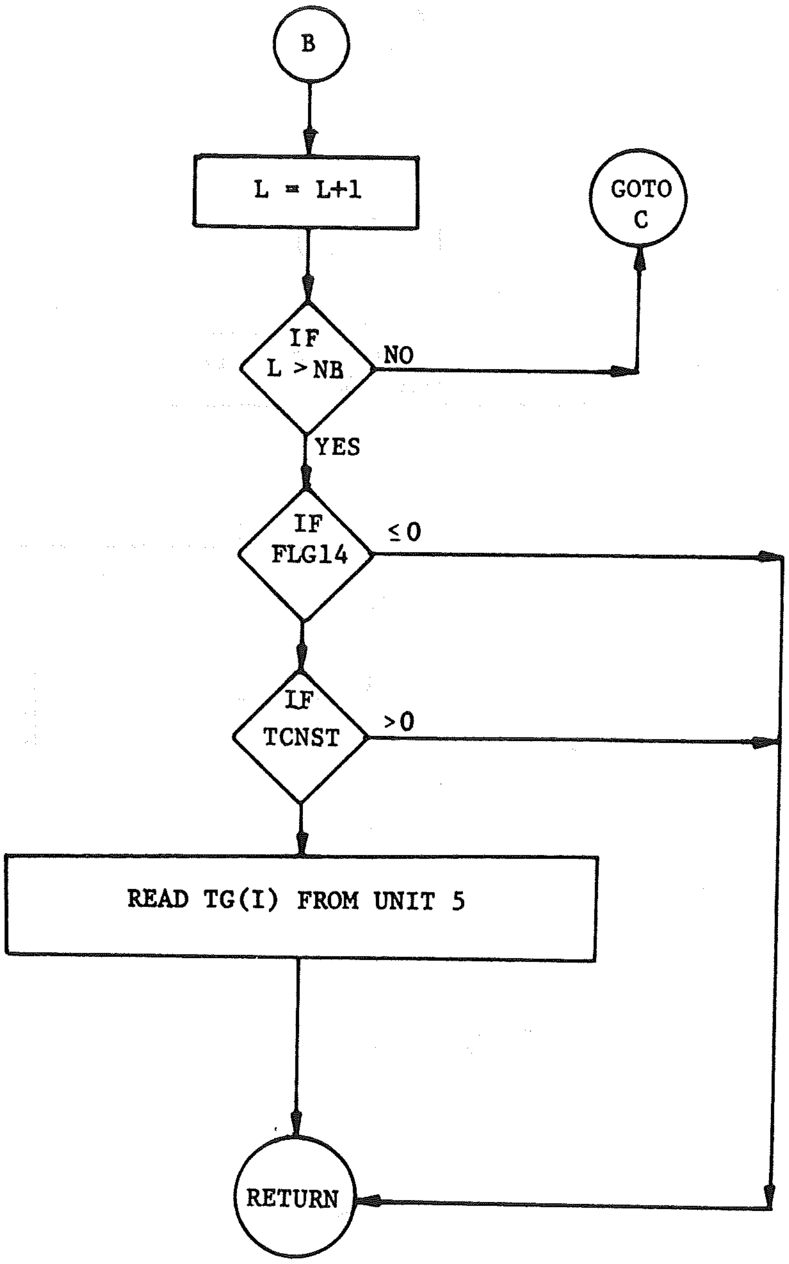


BASIC FLOW CHART FOR SUBROUTINE BASIC1

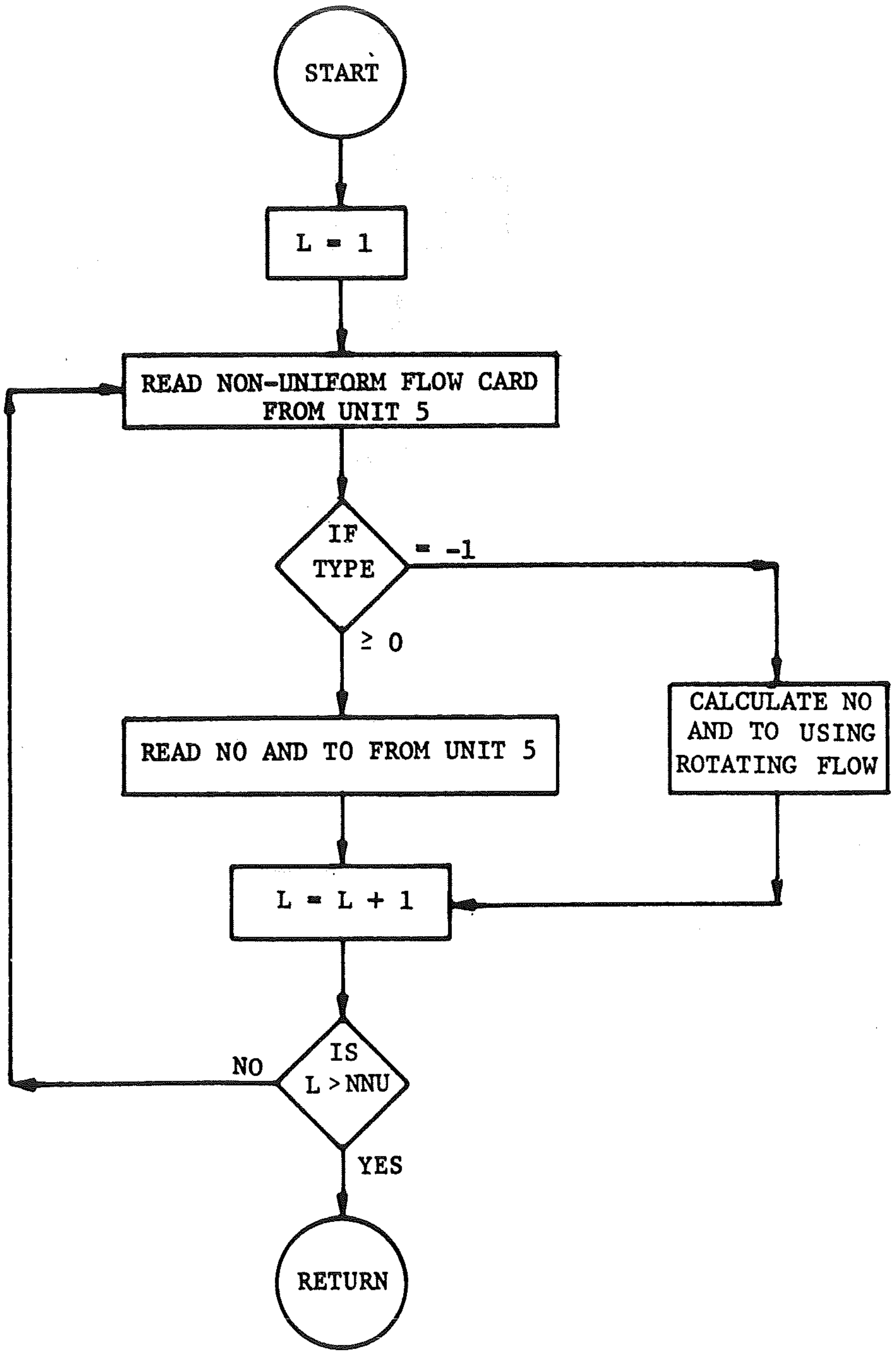

START
L=1
READ BODY TRANSFORMATION CARD FROM UNIT 5
C
READ BODY CONTROL CARD FROM UNIT 5
IF SUBKS
≠ 0
= 0
READ BODY COORDINATES FROM UNIT 13 (SAVED FROM PREVIOUS CASE)
IF FLG07
≠ 0
GENERATE ELLIPSE FROM XE AND YE INPUT DATA
READ BODY COORDINATES FROM UNIT NIN
GOTO A

BASIC FLOW CHART FOR SUBROUTINE BASIC1 (CONTINUED)

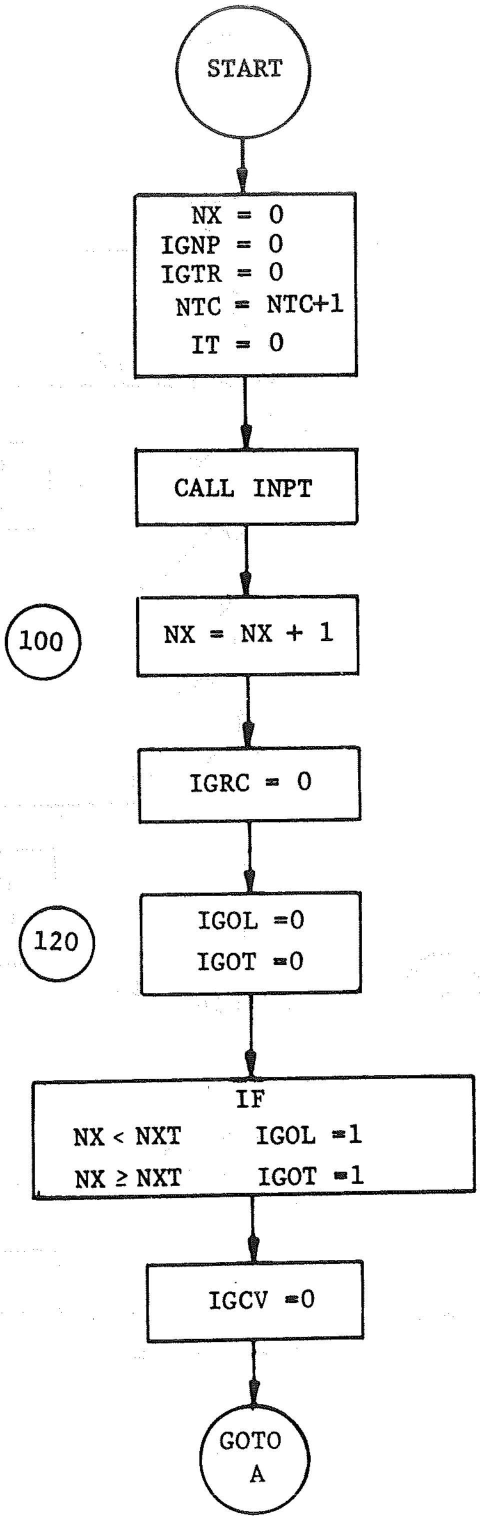



B
L = L+1
GOTO C
IF L > NB
NO
YES
IF FLG14
≤ 0
IF TCNST
> 0
READ TG(I) FROM UNIT 5
RETURN

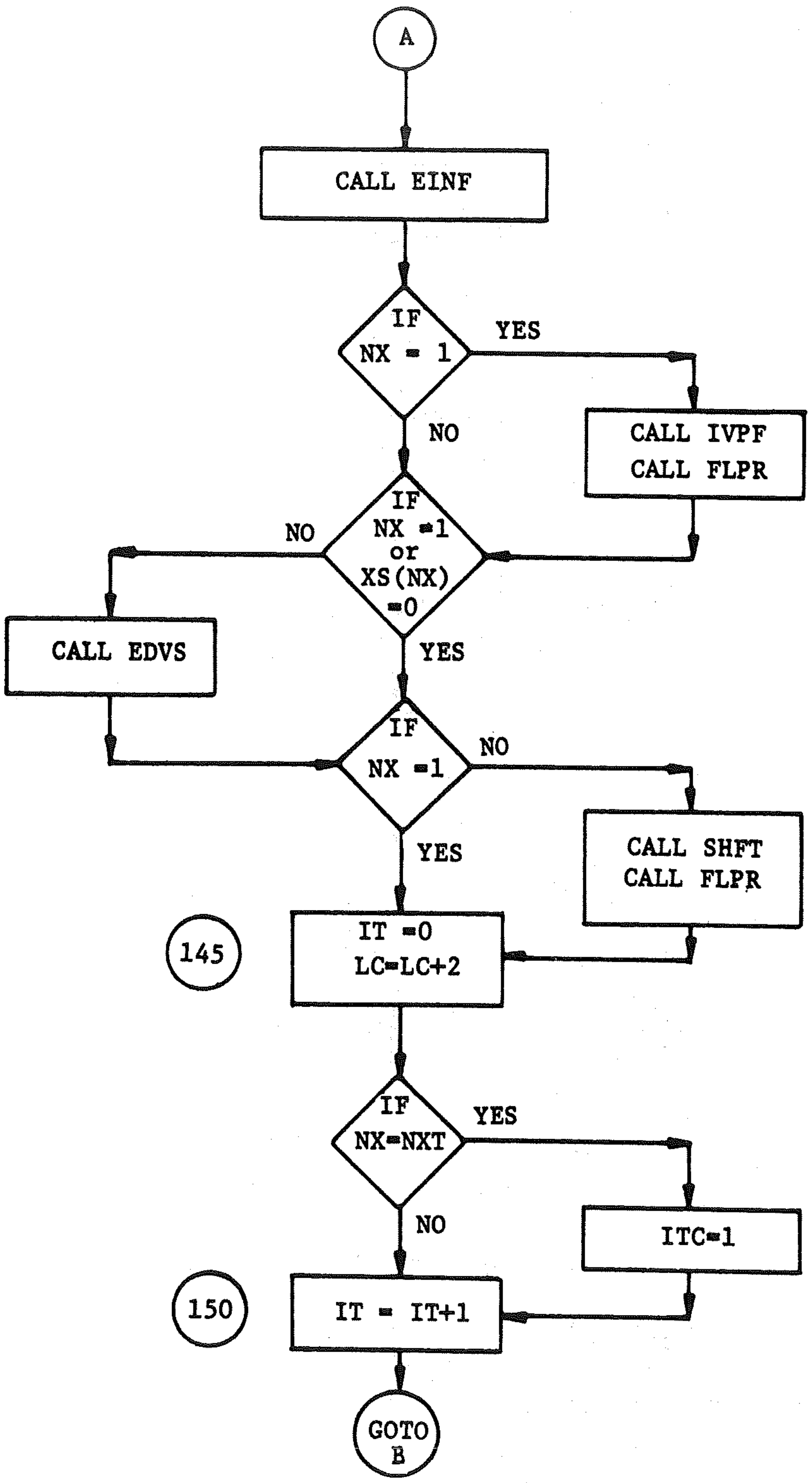
BASIC FLOW CHART FOR SUBROUTINE BASIC2


START
L = 1
READ NON-UNIFORM FLOW CARD FROM UNIT 5
IF TYPE
= -1
≥ 0
READ NO AND TO FROM UNIT 5
CALCULATE NO AND TO USING ROTATING FLOW
L = L + 1
IS L > NNU
NO
YES
RETURN

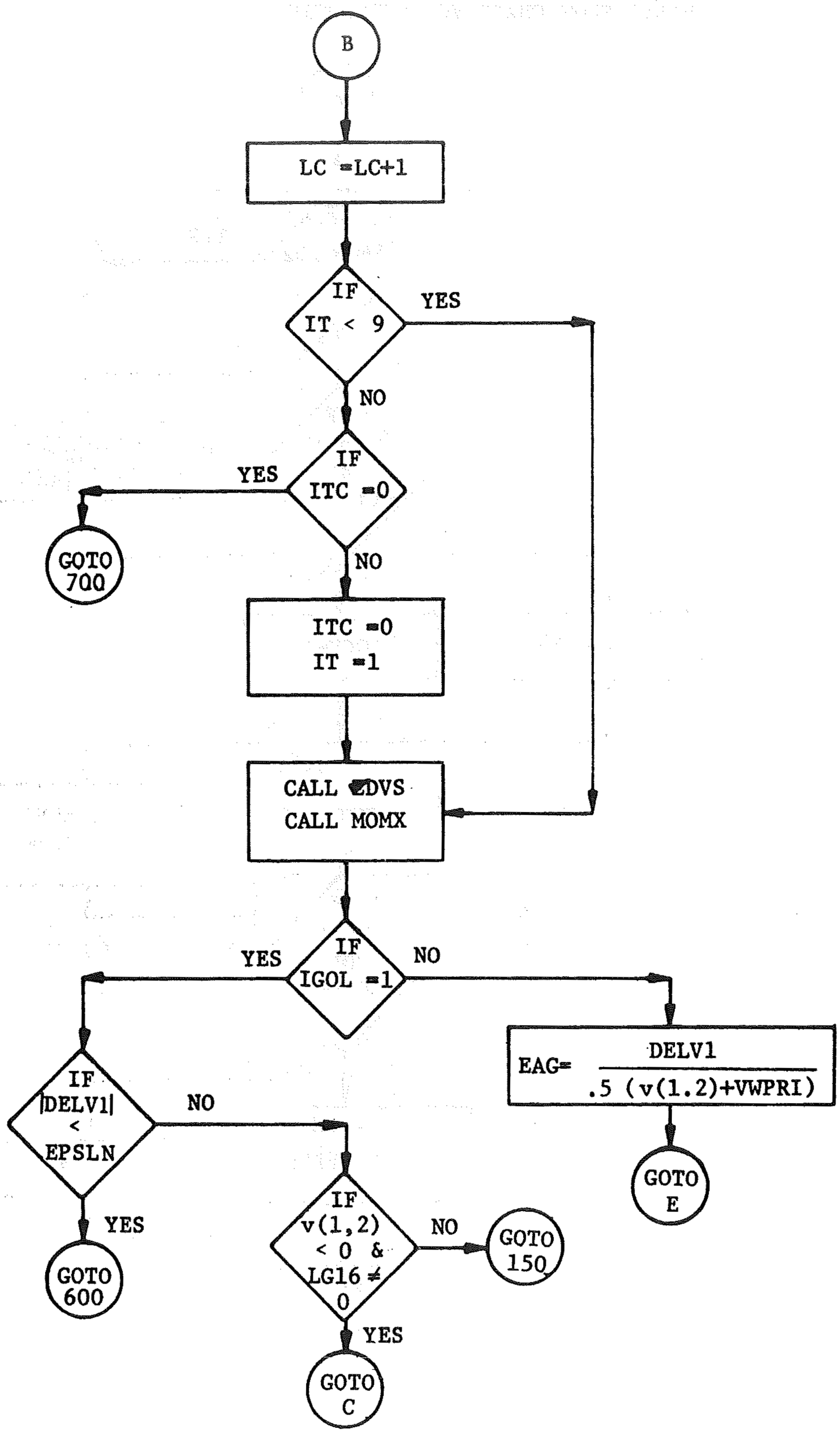
BASIC FLOW CHART FOR SUBROUTINE BOUNDL



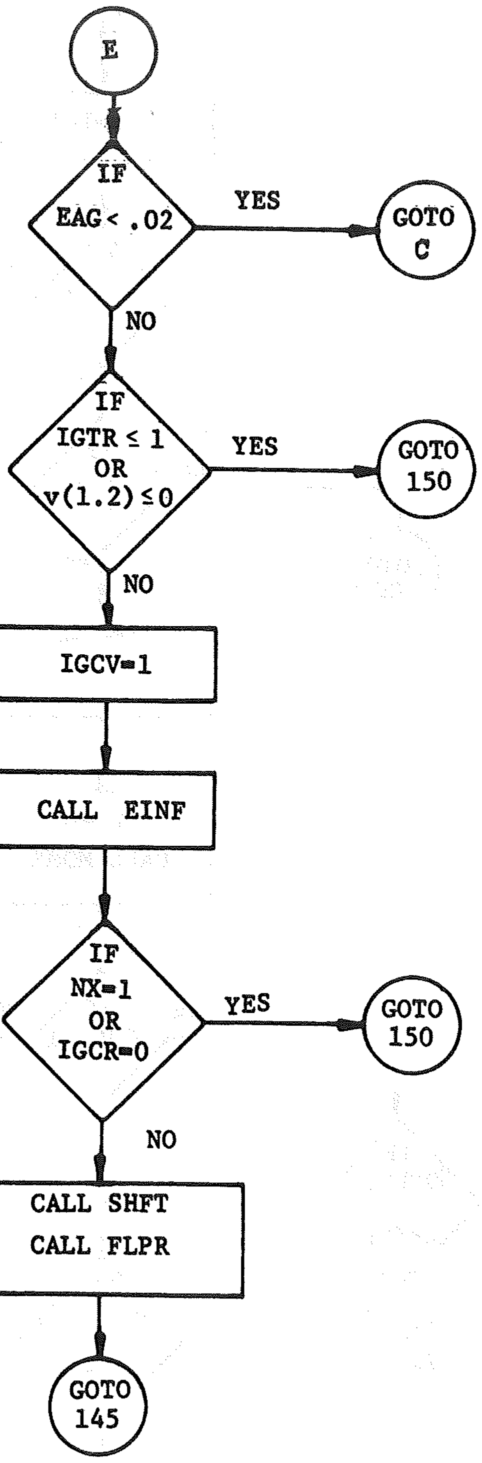
BASIC FLOW CHART FOR SUBROUTINE BOUNDL (CONTINUED)



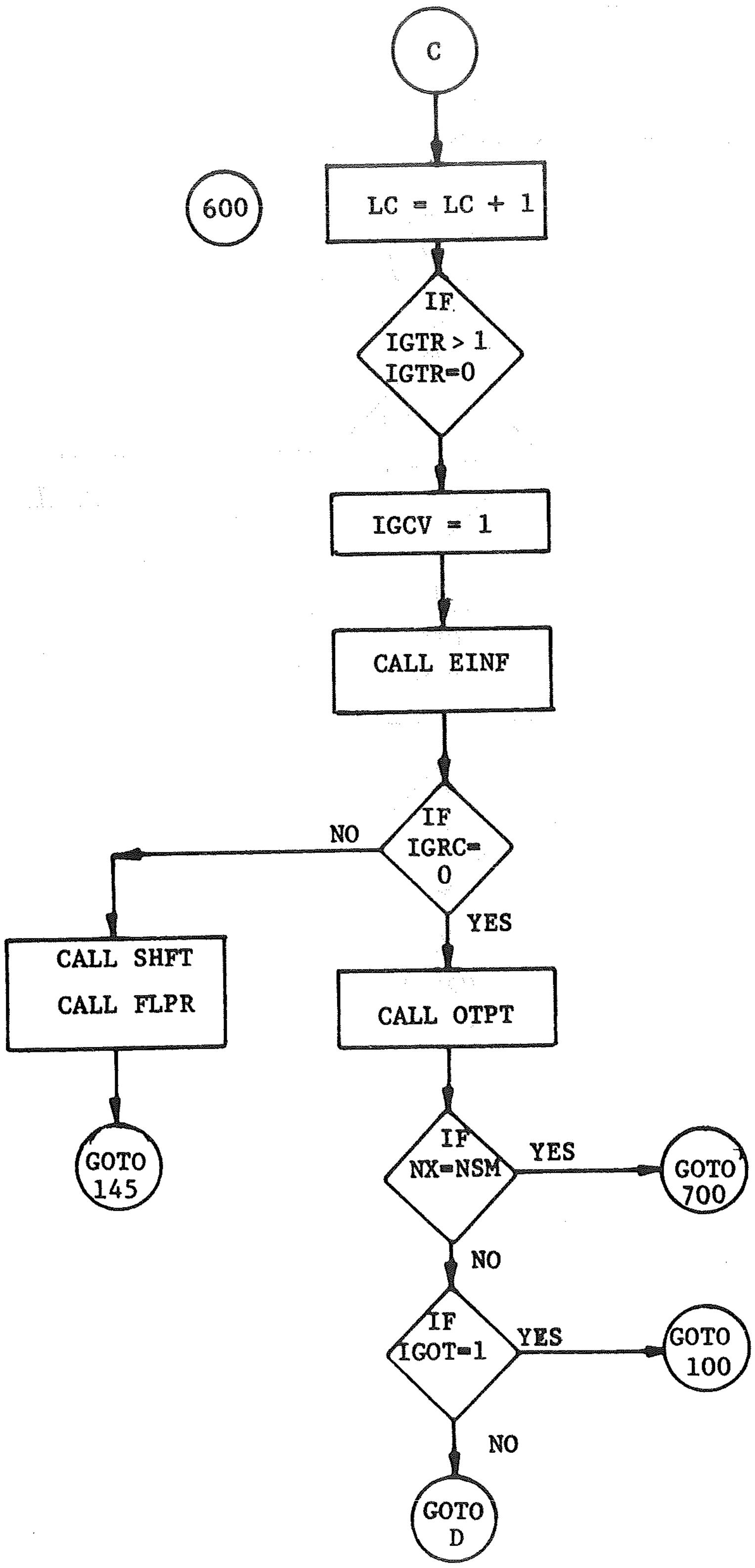
BASIC FLOW CHART FOR SUBROUTINE BOUNDL (CONTINUED)



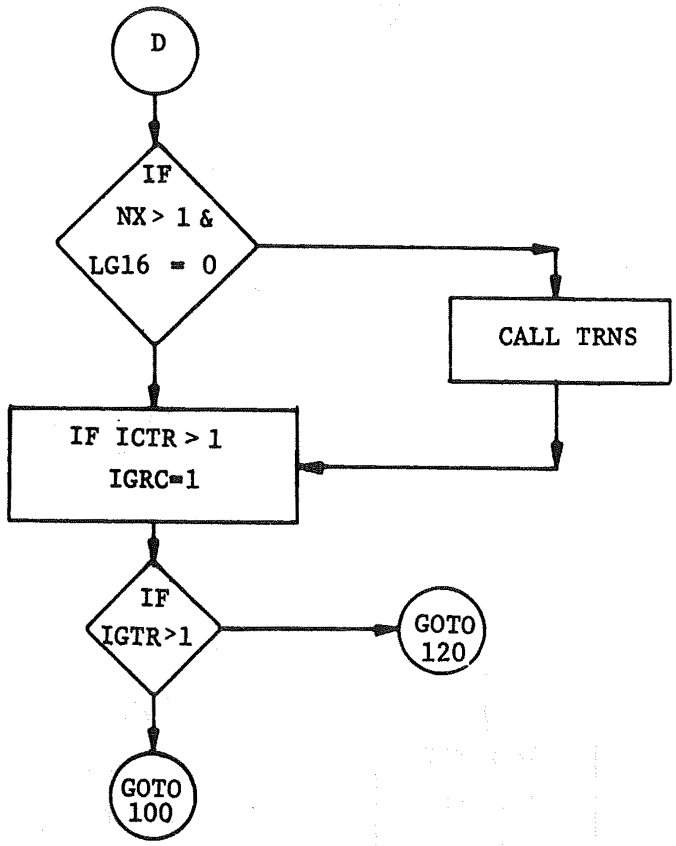
BASIC FLOW CHART FOR SUBROUTINE BOUNDL (CONTINUED)



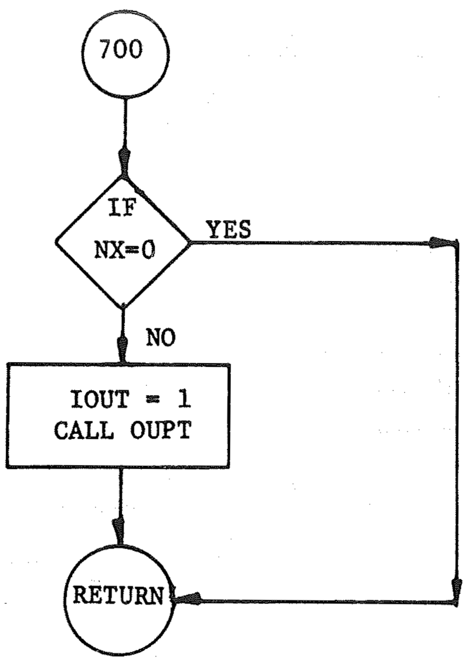
BASIC FLOW CHART FOR SUBROUTINE BOUNDL (CONTINUED)



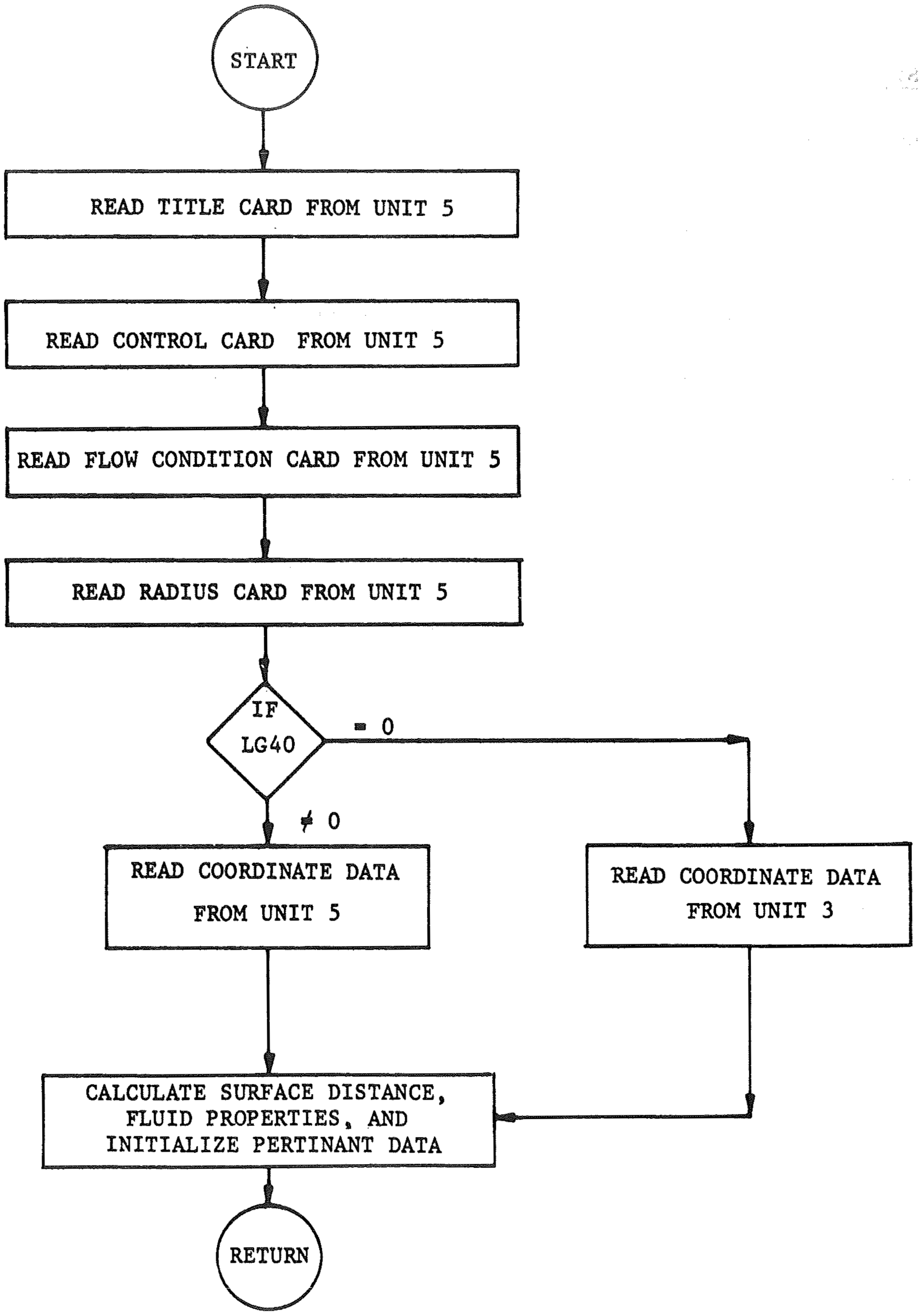
BASIC FLOW CHART FOR SUBROUTINE BOUNDL (CONTINUED)



BASIC FLOW CHART FOR SUBROUTINE BOUNDL (CONTINUED)



BASIC FLOW CHART FOR SUBROUTINE INPT


START
READ TITLE CARD FROM UNIT 5
READ CONTROL CARD FROM UNIT 5
READ FLOW CONDITION CARD FROM UNIT 5
READ RADIUS CARD FROM UNIT 5
IF LG40
= 0
≠ 0
READ COORDINATE DATA FROM UNIT 5
READ COORDINATE DATA FROM UNIT 3
CALCULATE SURFACE DISTANCE, FLUID PROPERTIES, AND INITIALIZE PERTINANT DATA
RETURN

The following is a description of the output symbols from the various sections of the ADAM computer program:

NEUMANN POTENTIAL FLOW SUBPROGRAM

SYMBOL	DESCRIPTION
ADDED MASS	$\Sigma 2\pi \cdot \text{Phi} \cdot V_n \cdot ds$
AJK	Influence coefficients W_{jk} resolved parallel to the outward normal of the element. Output only if FLG08 = 1.
ADDX	Constant which is added to all X-coordinates of a particular body. Value printed out for each body.
ADDY	Constant which is added to all Y-coordinates of a particular body. Value printed out for each body.
BJK	Influence coefficient W_{jk} resolved parallel to the tangent direction of the element.
BODIES	Number of bodies in system, same as NB input on flag card.
BODY NO.	Number of this particular body. This parameter input on body control card.
CHORD	The reference chord for the system.
COSA	The cosine of DALPHA
CP	The pressure coefficient on a body element.
DALPHA	The change in angle between consecutive elements of a body. (degrees)
DELTAS	The length of a body element.
MACH NO.	Mach number used in Gothert's transformation.
MX	The factor by which all X-coordinates are multiplied for one body. Input on body transformation card.
MY	The factor by which all Y-coordinates are multiplied for one body. Input on body transformation card.

SYMBOL	DESCRIPTION
N	Velocity normal to a surface element, this is a measure of how well the boundary condition of zero normal velocity is satisfied.
NN	The number of geometry data points for a given body. This is input on the body transformation card.
NNU	The number of non-uniform onset flows to be considered.
PHI	Value of potential on each surface element.
PSF NO.	Identification for this case..
SIGMA	Source density on each surface element.
SINA	Sin of DALPHA
SUM(T) DELTA(S)	This is the summation of T multiplied by Deltas up to each element midpoint.
SUMDS	Summation of Deltas, surface distance around the body.
TCNST	Constant value of tangential velocity used in special option.
THETA	The angle through which a body is to be rotated about the origin in a clockwise direction.
TI	The velocity at each midpoint.
VOLUME	The volume of the body being analyzed. (calculated by Neumann)
X	The input X-coordinate defining the body surface, or off-body X-coordinates.
XE	The value of semi-major axis used in ellipse generation option.
Y	The input Y-coordinates defining the body surface, or off-body Y-coordinates.
YE	The value of semi-minor axis used in ellipse generation option.

OUTPUT DATA SYMBOLS

Finite Difference Boundary Layer Subprogram

Output of this routine consists of CASE DATA and STATION DATA inputs as well as the computed STATION DATA. Body geometry data, flags and counters, and reference quantities are printed out under the heading of CASE DATA. Values of parameters at the outer edge of the boundary layer as well as the boundary condition inputs are printed out under the heading of STATION DATA. These are followed by iteration results, velocity profiles for each x-station (if FLG32 = 0), and a summary of the computed boundary-layer parameters as functions of streamwise or x-distance.

Error messages generated by the program are printed out at the end of the STATION DATA printout if they are generated by input errors. Other error messages are issued at different locations in the profile printout if errors are detected during the computations.

SYMBOL	DESCRIPTION
ALPH1	Local body slope dy/dx.
ALPH2	Not used in this program.
BETA	$\beta = (2\xi/u_e)(du_e/d\xi)$
C	CHORD
CDBASE	Base drag coefficient.
CF	$c_f = \tau_w/(1/2\ u_e^2)$, value of local skin friction coefficient.
CFA	Total integrated skin friction to each point.
C_P	Pressure coefficient
DELS	Boundary layer displacement thickness.
DELVW	Delta V(1,2) used in iteration for V(1,2).
EPS	ε^+, eddy viscosity parameter for outer region.
EPS1	ε_1, eddy viscosity parameter for inner region.
ETA	η, non-dimensionalized boundary layer thickness to each point in the boundary layer.

SYMBOL	DESCRIPTION
ETAE	η_∞ value of η which corresponds to δ.
ETAINF	Non-dimensional boundary layer thickness used as maximum. value in forming numerical solution grid.
F,FP,FPP	f,f',f", respectively.
FPPW	f" at the wall.
FPW	$f' = U/U_e$ at the wall.
FW	f_w, this is the transformed stream function at the wall. $$f_w = -\frac{1}{(2\xi)^{1/2}} \int_0^\xi \frac{V_w}{\mu_e u_e} d\xi$$
GW	Not used in this method.
GPW	Not used in this method.
H	Boundary layer form factor, $H = \delta^*/\theta$
HE	Enthalpy
H1	Initial step size, same as DETA1 in input.
IMAX	Number of points taken through the boundary layer.
K	Initial step size of the variable grid system.
KK	Variable grid parameter chosen internally.
ME	Local Mach number.
MUE	Local dynamic viscosity, μ_e, at edge of boundary layer.
MREF	Free stream Mach number.
MUREF	Free stream dynamic viscosity, μ_∞.
PE	Pressure at edge of boundary layer, P_e.
PRO	Laminar Prandtl number.
QW	Not used in this program.
REY	Reynolds number based on reference conditions (see input)

SYMBOL	DESCRIPTION
RHOREF	Freestream reference density.
R_o/C	R_o/L axisymmetric radius.
RR	Not used by this program.
RTHETA	Momentum thickness Reynolds number, R_θ.
R_x	Reynolds number based on local conditions. $$R_x = \frac{u_e x}{\nu}$$
S	Surface distance.
S/C	Nondimensionalized surface distance.
SHORTP	Flag which tells program to print velocity profiles. Same as FLG32 in input.
SQUIG	Transformed x-coordinate, ξ $$\xi = \int_0^x \rho_e \mu_e u_e \left(\frac{r_o}{L}\right)^{2k} dx$$
ST	Not used in this program.
SWEEP	Not used in this program.
TE	Temperature through boundary layer. Not needed for this program.
THETA	Momentum thickness, θ.
TREF	Reference temperature, T_∞.
TRFLAG	Flag which determines transition (input).
TRINT	Flag which determines instantaneous transition or use of transitional region option (input).
TW	Temperature at the wall. Not used in this program.
TVC	Transverse curvature flag (input).
UE	Velocity at edge of boundary layer.
UPLUS	Non-dimensionlized velocity in the boundary layer.

SYMBOL	DESCRIPTION
X	X-coordinate
X/C	Non-dimensionalized x-coordinate
XI	Transformed x-coordinate - same as SQUIG
Y	Y-coordinate
Y/C	Non-dimensionalized y-coordinate
YPLUS	Non-dimensionalized y-coordinate in boundary layer.
VREF	Reference velocity (input)

Iteration Subprogram

XNEW and YNEW These are the coordinates of the equivalent viscous body. The original coordinates modified by the addition of the boundary layer displacement thickness δ^*.

DESCRIPTION OF STORAGE UNITS

The following is a description of all disk storage units used in ADAM:

TAPE	DESCRIPTION
TAPE1	This unit used in subroutine SOLVIT as a scratch unit and also used to transfer the "viscous" coordinates from subroutine iterat to subroutine smooth or subroutine BASIC1.
TAPE2	This unit used in subroutine SOLVIT as a scratch unit and also used to transfer the boundary layer displacement thickness's from subroutine OTPT to subroutine ITERAT.
TAPE3	This unit used to store source densities in subroutine SOLVIT and used to transfer body geometry and pressures from subroutine AXIS to subroutine INPT.
TAPE4	sin α, cos α, TCNST, TG(I), cos R^2, $2\|(\sin\alpha)(\cos\alpha)\|$, N_o, T_o, V_N, T_T, A(J), B(J), etc. Used exclusively in Neumann subprogram to store and transfer data.
TAPE5	This tape used for card input.
TAPE6	This tape used for printed output.
TAPE8	This tape used to store extra cross flow matrices, EC , ECY, ECZ, in subroutine MATRIX.
TAPE9	This tape used to store axisymmetric flow matrices AS, AY, AZ, in subroutine MATRIX.
TAPE10	This tape used to store cross flow matrices CX, CY, CZ in subroutine matrix and also used to transfer smoothed

TAPE	DESCRIPTION
TAPE10 (Continued)	coordinate data from subroutine smooth to subroutine BASIC1.
TAPE11	This tape used as a scratch unit in subroutine SOLVIT.
TAPE12	This tape used to store transformed parameters X1, Y1, X2, Y2, and ΔS_i.
TAPE13	This tape used to store untransformed coordinates (TX1, TY1) for use in SUBCASE option.
TAPE15	This tape used to store transformed coordinates, (X1, Y1) for use in subroutine ITERAT.

APPENDIX C

PROGRAM LISTINGS

This part of the report contains the source card listings for the axisymmetric design and analysis method (ADAM) computer program. This program may be run either on a CDC or an IBM computer. The listing as presented here is for the CDC version of the program. This program has been run on the CDC 6600 computer. The program is written in FORTRAN for the CDC run compiler and has been run under the scope 3.1 and 3.4 operating systems. In this listing all cards that are peculiar to the CDC version of FORTRAN are identified by a C in card column 80. All cards that are peculiar to the IBM FORTRAN IV compiler are identified by an I in card column 80 and a C in card column 1. In other words, the code for both CDC and IBM machines is in the deck but the IBM cards are made inactive by converting them to comment statements (C in card column 1). Since all of the machine dependent cards are identified by an I or C in card column 80 it is a simple matter to convert the deck from one version to the other with a small conversion program. When converting from CDC to IBM code this conversion program reads and copies each card to a storage unit. If a card has a C in card column 80, then a C is written into card column 1 to make the CDC peculiar card inactive. If a card has an I in card bolumn 80, then the C is removed from card column 1 and the card image written to the storage unit as an active FORTRAN statement. The conversion from IBM back to CDC is made in a similar manner. The conversion program to convert the deck from CDC to IBM FORTRAN is listed below (for use on an IBM machine):

```
      DIMENSION DATA(22)
      DATA CB,CC,CI/1H, 1HC,1HI/
      REWIND 19
      DO 100 I=1,20000
        READ (5,20,END=300) DATA
 20       FORMAT (1A1,19A4,1A2,1A1)
      IF (DATA(22) .EQ. CI) DATA(1) = CB
      IF (DATA(22) .EQ. CC) DATA(1) = CC
      WRITE (19,20) DATA
100   CONTINUE
300   STOP
      END
```

This program places the new deck with IBM cards made active, and CDC cards inactive, on to unit 19. The references to unit 19 above can be changed to unit 7 to punch the deck out.

```
      OVERLAY(AXSY,0,0)                                                 MAIN  001C
      PROGRAM MAIN(INPUT=201,OUTPUT,          TAPE5=INPUT,TAPE6=OUTPUT, MAIN  002C
     1              TAPE1=201,TAPE2=201,TAPE3=201,TAPE4=201,TAPE8=201,  MAIN  003C
     2 TAPE9=201,TAPE10=201,TAPE11=201,TAPE12=201,TAPE13=201,TAPE15=201)MAIN  004C
C     B8AC MAIN PROGRAM                                                 MAIN  005
C                                                                       MAIN  006
C     THIS IS THE AXISYMMETRIC DESIGN AND ANALYSIS METHOD ,ADAM,        MAIN  007
C     COMPUTER PROGRAM.  THIS COMPUTER PROGRAM WILL CALCULATE THE       MAIN  008
C     AERODYNAMIC FORCES ACTING ON AN AXISYMMETRIC BODY OPERATING       MAIN  009
C     IN A VISCOUS INCOMPRESSIBLF FLOW FIELD                            MAIN  010
C                                                                       MAIN  011
    1 FORMAT(5I4)                                                       MAIN  012
C                                                                       MAIN  013
    2 READ(5,1) IGEOM,INEUM,IBOUND,ITER,IFINSH                          MAIN  014
C                                                                       MAIN  015
C     THESE FOUR FLAGS DETERMINE WHICH ROUTINES WILL BE USED            MAIN  016
C                                                                       MAIN  017
C     IGEOM CONTROLS THE GEOMETRY DEFINITION                            MAIN  018
C     IF  IGEOM = 0  NO SMOOTHING IS USED                               MAIN  019
C     IF  IGEOM = 1     SMOOTHING IS USED                               MAIN  020
C                                                                       MAIN  021
C     INEUM INDICATES WHETHER OR NOT A POTENTIAL FLOW SOLUTION WILL     MAIN  022
C     BE GENERATED                                                      MAIN  023
C     IF  INEUM = 0  NO POTENTIAL FLOW SOLUTION IS USED                 MAIN  024
C     IF  INEUM = 1  A  POTENTIAL FLOW SOLUTION IS USED                 MAIN  025
C                                                                       MAIN  026
C     IBOUND INDICATES WHETHER OR NOT A BOUNDARY LAYER SOLUTION IS      MAIN  027
C     DESIRED                                                           MAIN  028
C     IF IBOUND = 0  NO BOUNDARY LAYFR SOLUTION IS NEEDED               MAIN  029
C     IF IBOUND = 1  A BOUNDARY LAYER SOLUTION IS NEEDED                MAIN  030
C                                                                       MAIN  031
C     ITER CONTROLS THE ITERATION CYCLE                                 MAIN  032
C     IF ITER = 0  NO ITERATION IS NEEDED                               MAIN  033
C     IF ITER = 1  AN ITERATION IS NEEDED                               MAIN  034
C                                                                       MAIN  035
```

```
      IGFOM = IGEOM + 1                                                 MAIN  036
      INFUM = INFUM + 1                                                 MAIN  037
      TBOUND = IBOUND + 1                                               MAIN  038
      ITFR = ITER + 1                                                   MAIN  039
C                                                                       MAIN  040
      GO TO (30,20), IGEOM                                              MAIN  041
C                                                                       MAIN  042
C  20 CALL SMOOTH                                                       MAIN  043I
   20 CALL OVERLAY(4HAXSY,6,0,6HRECALL)                                 MAIN  044C
C                                                                       MAIN  045
   30 GO TO (60,50),INEUM                                               MAIN  046
C                                                                       MAIN  047
   50 CALL NEUMAN                                                       MAIN  048
C                                                                       MAIN  049
   60 GO TO (90,80), IBOUND                                             MAIN  050
C                                                                       MAIN  051
C  80 CALL BOUNDL                                                       MAIN  052I
   80 CALL OVERLAY(4HAXSY,5,0,6HRECALL)                                 MAIN  053C
C                                                                       MAIN  054
   90 GO TO (120,110), ITER                                             MAIN  055
C                                                                       MAIN  056
C 110 CALL ITERAT                                                       MAIN  057I
  110 CALL OVERLAY(4HAXSY,7,0,6HRECALL)                                 MAIN  058C
C                                                                       MAIN  059
  120 IF (IFINSH .NE. 9999) GO TO 2                                     MAIN  060
C                                                                       MAIN  061
C     STOP                                                              MAIN  062I
  200 CONTINUE                                                          MAIN  063C
      FND                                                               MAIN  064
      SUBROUTINE NEUMAN
C*           ** DOUGLAS NEUMANN POTENTIAL FLOW PROGRAM **
C
C             * CALCULATION OF POTENTIAL FLOW ABOUT BODIES OF
C               REVOLUTION HAVING FLOWS PARALLEL AND PERPENDICULAR
C               TO THE AXIS OF REVOLUTION.
C
C             * MAIN PROGRAM
C
      COMMON /IPSF/ PSF
       COMMON / NBSAVE / NBOLD, NIN
      COMMON      HEDR(10)  ,CASE      ,NB        ,NNU
     1           ,FLG03     ,FLG04     ,FLG05     ,FLG06     ,FLG07
     2           ,FLG08     ,FLG09     ,FLG10     ,FLG11     ,FLG12
     3           ,FLG13     ,FLG14     ,FLG15     ,FLG16     ,FLG17
     4           ,FLG18     ,FLG19     ,FLG20     ,FLG21     ,FLG22
     5           ,FLG23     ,FLG24     ,FLG25     ,FLG26     ,FLG27
      COMMON   NT,       ND(11),   MN,       NUNA(5),  TYPEA(5),
     1         NER1,     NER2,     NMA,      NSIGA,    NSIGC,
     2         NUNC(5),  TYPEC(5), NLF(11),  IFC,      NSIGEC,
     3         TYPFEC(5),NUNEC(5)
      COMMON/ITERF/ ITER
C     DOUBLE PRECISION  HEDR, CASE
      INTEGER      FLG03     ,FLG04     ,FLG05     ,FLG06     ,FLG07
     1            ,FLG08     ,FLG09     ,FLG10     ,FLG11     ,FLG12
     2            ,FLG13     ,FLG14     ,FLG15     ,FLG16     ,FLG17
     3            ,FLG18     ,FLG19     ,FLG20     ,FLG21     ,FLG22
     4            ,FLG23     ,FLG24     ,FLG25     ,FLG26     ,FLG27
      REAL         MN
       NBOLD = 0
C
      REWIND 3
      REWIND 4
      REWIND 8
      REWIND 9
      REWIND 10                                                  NEUM  036
      REWIND 11                                                  NEUM  037
   10 REWIND 12                                                  NEUM  038
      REWIND 13                                                  NEUM  039
      REWIND 15                                                  NEUM  040
C     CALL PART1                                                 NEUM  041I
      CALL OVERLAY (4HAXSY,1,0,6HRECALL )                        NEUM  042C
      IF(FLG06 .NE. 0) GO TO 50                                  NEUM  043
C  30 CALL PREP                                                  NEUM  044I
   30 CALL OVERLAY (4HAXSY,3,0,6HRECALL )                        NEUM  045C
C  40 CALL PART4                                                 NEUM  046I
   40 CALL OVERLAY (4HAXSY,4,0,6HRECALL )                        NEUM  047C
   50 RETURN                                                     NEUM  048
      END                                                        NEUM  049
      SUBROUTINE INS1 ( ARG, TABLE, OTPT  , NLQ, NFR )
C     INS1 - SINGLE LINEAR OR QUADRATIC INTERPOLATION
C              ONE OR TWO FUNCTIONS OF ONE VARIABLE
C
C     THIS SUBROUTINE WILL INTERPOLATE FOR EITHER
C        1)   F(X) FROM A TABLE OF X VRS F(X), OR
C        2)   F(X) AND G(X) FROM A TABLE OF X VRS F(X),G(X).
C        THE TABLE MAY HAVE UNEQUAL SPACING IN X.  EITHER LINEAR
C        OR QUADRATIC LAGRANGE INTERPOLATION MAY BE USED.
C
C
C        ARG    = INPUT  = X ARGUMENT
C
C        TABLE  = INPUT  = IS A LINEAR ARRAY.  THE FIRST WORD IS AN
C                          INTEGER CODE (EITHER INTEGER FORM OR REAL*4
C                          FORM).  IF THIS CODE IS POSITIVE, THE CODE
C                          SPECIFIES THE NUMBER OF X,F(X) PAIRS
C                          IMMEDIATELY FOLLOWING THE CODE IN SUCCESSIVE
C                          WORDS.  IF THE CODE IS NEGATIVE,
C                          ABSOLUTE VALUE OF THE CODE SPECIFIES THE
C                          NUMBER OF X,F(X),G(X) TRIPLES IMMEDIATELY
C                          FOLLOWING THE CODE IN SUCCESSIVE WORDS.
C                          THE X-VALUES MUST BE IN ASCENDING ORDER.
C                          EXCEPT FOR THE CODE, THE INPUT TABLE VALUES
C                          MUST BE IN REAL*4 FORM
C
C        OTPT   = OTPT   = INTERPOLATED VALUE OF F(X) IF TABLE(1) IS
C                                POSITIVE
C                        = A TWO WORD ARRAY (IF TABLE(1) IS NEGATIVE)
C                                CONTAING THE INTERPOLATED VALUES
C                                FOR F(X) AND G(X)
C
C        NLQ    = INPUT  = INTERPOLATION FLAG (INTEGER)
C                        = 1  FOR LINEAR INTERPOLATION
C                        = 2  FOR QUADRATIC INTERPOLATION
```

```
C                                                                        INS1  036
C        NER    = OUTPUT = ERROR CODE (INTEGER)                          INS1  037
C                        = 1  INTERPOLATION SUCCESSFUL                   INS1  038
C                        = 2  BELOW TABLE, MINIMUM VALUE FURNISHED       INS1  039
C                        = 3  ABOVE TABLE, MAXIMUM VALUE FURNISHED       INS1  040
C                        = 4  NOT ENOUGH ENTRIES - NO ANSWER             INS1  041
C                        = 5  X-VALUES NOT IN ASCENDING ORDER - NO       INS1  042
C                             ANSWERS                                    INS1  043
C                                                                        INS1  044
      DIMENSION OTPT(2), TABLE(1)                                        INS1  045
C                                                                        INS1  046
      EQUIVALENCE  ( NOENTR, A )                                         INS1  047
C                                                                        INS1  048
      A = ABS( TABLE(1) )                                                INS1  049
      M = 2                                                              INS1  050
      IF  ( TABLE(1) .LT. 0.0 )  M = 3                                   INS1  051
      MM1 = M - 1                                                        INS1  052
      MP1 = M + 1                                                        INS1  053
      IF  ( A .GT. 0.99E0 )  NOENTR = A + 0.5E0                          INS1  054
      J = 3                                                              INS1  055
C                                                                        INS1  056
C     CHECK FOR NUMBER OF ENTRIES                                        INS1  057
C                                                                        INS1  058
      IF  ( NLQ .NE. 1 )  GO TO 20                                       INS1  059
      IF  ( NOENTR .GE. 2 )  GO TO 30                                    INS1  060
   10 NER = 4                                                            INS1  061
      GO TO 140                                                          INS1  062
   20 IF  ( NOENTR .LT. 3 )  GO TO 10                                    INS1  063
C                                                                        INS1  064
C     CHECK FOR ARGUMENT LESS THAN OR EQUAL TO LOW LIMIT                 INS1  065
C                                                                        INS1  066
   30 IF  ( ARG - TABLE(2) )40,50,60                                     INS1  067
   40 NER = 2                                                            INS1  068
      GO TO 130                                                          INS1  069
   50 NER = 1                                                            INS1  070
```

```
      GO TO 130
C
C     SEARCH FOR ENTRY WITHIN TABLE
C
   60 NOS = M * NOFNTR
      IST = 2 + M
      DO  90  I = IST, NOS, M
      J = I + 1
      IF  ( TABLE(I) - TABLE(I-M) )70,70,80
   70 NER = 5
      GO TO 140
   80 IF  ( TABLE(I) - ARG )90,50,100
   90 CONTINUE
      NER = 3
      GO TO 130
C
C     SEARCH SUCCESSFUL, TEST INTERPOLATION TYPE
C
  100 IF  ( NLQ .GT. 1)  GO TO 110
C
C     USE LINEAR INTERPOLATION
C
      NER = 1
      OTPT  (1) = TABLE(I+1) * ( ARG - TABLE(I-M) ) / ( TABLE(I) -
     1            TABLE(I-M) ) + TABLE(I-MM1) * ( ARG - TABLE(I) ) /
     2            ( TABLE(I-M) - TABLE(I) )
      IF  ( M .NE. 3 )  GO TO 140
      OTPT  (2) = TABLF(I+2) * ( ARG - TABLE(I-7) ) / ( TABLE(I) -
     1            TABLE(I-7) ) + TABLE(I-1) * ( ARG - TABLF(I) ) /
     2            ( TABLE(I-M) - TABLE(I) )
      GO TO 140
C
C     USE QUADRATIC INTERPOLATION
C
  110 I = I - M
```

```
      IF  ( I .GE. 2*M )  GO TO 120                                    INS1  106
      I = I + M                                                        INS1  107
  120 XA1 = ARG - TABLE(I)                                             INS1  108
      XA0 = ARG - TABLE(I-M)                                           INS1  109
      XA2 = ARG - TABLE(I+M)                                           INS1  110
      X01 = TABLE(I-M) - TABLE(I)                                      INS1  111
      X02 = TABLE(I-M) - TABLE(I+M)                                    INS1  112
      X12 = TABLE(I)   - TABLE(I+M)                                    INS1  113
      NER = 1                                                          INS1  114
      OTPT  (1) = TABLE(I-MM1) * ( XA1 / X01 ) * ( XA2 / X02 ) -       INS1  115
     1            TABLE(I+1)   *  ( XA0 / X01 ) * ( XA2 / X12 ) +      INS1  116
     2            TABLE(I+MP1) * ( XA0 / X02 ) * ( XA1 / X12 )         INS1  117
      IF  ( M .NE. 3 )  GO TO 140                                      INS1  118
      OTPT  (2) = TABLE(I-1) * ( XA1 / X01 ) * ( XA2 / X02) -          INS1  119
     1            TABLE(I+2) * ( XA0 / X01 ) * ( XA2 / X12 ) +         INS1  120
     2            TABLE(I+5) * ( XA0 / X02 ) * ( XA1 / X12 )           INS1  121
      GO TO 140                                                        INS1  122
C                                                                      INS1  123
C     ERROR EXIT  -  SET OUTPUT VALUE                                  INS1  124
C                                                                      INS1  125
  130 OTPT  (1) = TABLE(J)                                             INS1  126
      IF  ( M .EQ. 3 )  OTPT  (2) = TABLE(J+1)                         INS1  127
C                                                                      INS1  128
C     NORMAL EXIT                                                      INS1  129
C                                                                      INS1  130
  140 RETURN                                                           INS1  131
      END                                                              INS1  132
      FUNCTION ARSIN(X)                                                 ARSN  001C
C   THIS ROUTINE IS REQUIRED BECAUSF OF DIFFERENCES BETWEEN CDC AND IBM  ARSN  002
C   FORTRAN.                                                            ARSN  003
      ARSIN = ASIN(X)                                                   ARSN  004C
      RETURN                                                            ARSN  005C
      END                                                               ARSN  006C
      OVERLAY(AXSY,1,0)                                                 PAR1  001C
      PROGRAM PART1                                                     PAR1  002C
C     SUBROUTINE PART1                                                  PAR1  003I
C                                                                       PAR1  004
C               * CONTROL FOR BASIC DATA AND FORM MATRIX                PAR1  005
C                                                                       PAR1  006
       COMMON / NBSAVE /  NBOLD, NIN                                    PAR1  007
      COMMON  /RNGWNG/ VA(100,2),  VR(100,2);VAN(100), VAT(100)         PAR1  008
      COMMON /ECF/ ECX(100), ECY(100), ECZ(100)                         PAR1  009
      COMMON  /D/ D1, D3, XMXJ, YMYJ, XMXJP1; YMYJP1, S                 PAR1  010
      COMMON       HEDR(10)  ,CASE       ,NB         ,NNU               PAR1  011
     1            ,FLG03     ,FLG04      ,FLG05      ,FLG06     ,FLG07  PAR1  012
     2            ,FLG08     ,FLG09      ,FLG10      ,FLG11     ,FLG12  PAR1  013
     3            ,FLG13     ,FLG14      ,FLG15      ,FLG16     ,FLG17  PAR1  014
     4            ,FLG18     ,FLG19      ,FLG20      ,FLG21     ,FLG22  PAR1  015
     5            ,FLG23     ,FLG24      ,FLG25      ,FLG26     ,FLG27  PAR1  016
      COMMON   NT,        ND(11),    MN,        NUNA(5),  TYPEA(5),     PAR1  017
     1         NER1,      NER2,      NMA,       NSIGA,    NSIGC,        PAR1  018
     2         NUNC(5),   TYPEC(5),  NLF(11),   IFC,      NSIGEC,       PAR1  019
     3         TYPEEC(5),NUNEC(5)                                       PAR1  020
C     DOUBLE PRECISION  HEDR, CASE                                      PAR1  021I
      INTEGER      FLG03     ,FLG04      ,FLG05      ,FLG06     ,FLG07  PAR1  022
     1            ,FLG08     ,FLG09      ,FLG10      ,FLG11     ,FLG12  PAR1  023
     2            ,FLG13     ,FLG14      ,FLG15      ,FLG16     ,FLG17  PAR1  024
     3            ,FLG18     ,FLG19      ,FLG20      ,FLG21     ,FLG22  PAR1  025
     4            ,FLG23     ,FLG24      ,FLG25      ,FLG26     ,FLG27  PAR1  026
      COMMON /IPSF/ PSF                                                 PAR1  027
      REAL         MN                                                   PAR1  028
C                                                                       PAR1  029
      COMMON /CL/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),  PAR1  030
     1              SINA(100),COSA(100),XP(100),  YP(100)               PAR1  031
     2             ,XWAKE(11),YWAKE(11)                                 PAR1  032
      COMMON /TL/   TX1(100), TY1(100), NG(100),  TG(100),  ALFA(100),  PAR1  033
     1  RSDS(100),DALF(100),CHORD,TCNST,DUMMY(1315)                     PAR1  034
      COMMON/ITERF/ ITER                                                PAR1  035
      INTFGER     BDN       ,SUBKS
      REAL        MX        ,MY       ,NG
C
C               * START
C               * RFAD INPUT DATA
  100 READ (5,4) HEDR, CASE, PSF , NB, NNU, FLG03, FLG04, FLG05, FLG06,
     1           FLG07, FLG08, FLG09, FLG10, FLG11, FLG12, FLG13, FLG14,
     2           FLG15, FLG16, FLG17, FLG18, FLG19, FLG20, FLG21, FLG22,
     3           FLG23, FLG24, FLG25, FLG26, FLG27,       NIN,ITER
C*** ***TRIANGULARIZATION OF THE MATRIX (SOLVIT) IS THE DEFAULT SOLUTION
      IF(FLG09.EQ.0.AND.FLG10.EQ.0)FLG13=1
C***   ***FLG22 IS GENERATED (RESEP) BOUNDARY CONDITIONS
C***   ***FLG21 IS EXTRA CROSS FLOW
    1 IF (FLG22.LE.0)GO TO 5
      FLG21 = 1
      FLG03 = 1
      FLG04 = 1
C***   ***IF FLAG 18 IS NOT EQUAL TO FLAG 14 YOU MUST USE DIRECT MATRIX
    5 IF (FLG18.NE.FLG14)GO TO 2
      IF (FLG21.LE.0)GO TO 3
      FLG12 = 1
    2 FLG13 = 1
      FLG09 = 0
      FLG10 = 0
    3 CONTINUE
        IF( NBOLD .EQ. 0 )       NBOLD = NB
C*** ***CARDS (UNIT 5) ARE THE DEFAULT METHOD OF INPUT
        IF( NIN .EQ. 0 )       NIN = 10
    4 FORMAT ( 10A6, 2X A6, 8X A4/ 27I1, I2,I1)
      READ (5,8) CHORD, MN, TCNST
    8 FORMAT ( 3F10.0 )
C*** ***THE DEFAULT CHORD LENGTH IS 1.0
      IF(CHORD.GT.-1.0E-5.AND.CHORD.LT.1.0E-5)CHORD=1.0
      WRITE (6,12) HEDR, CASE, NB, NNU, CHORD, MN, TCNST, PSF
   12 FORMAT ( 1H1 25X, 26HDOUGLAS  AIRCRAFT COMPANY /
     1          28X, 21HLONG  BEACH  DIVISION ///                      PAR1  071
     2           6X, 43HPROGRAM EODA -- AXISYMMETRIC AND CROSSFLOW //  PAR1  072
     3          11X, 29H***** CASE CONTROL DATA ***** ///              PAR1  073
     4           6X, 10A6, 4X, 10HCASE NO.  A6 //                      PAR1  074
     5           6X  9HBODIES  =I3/ 6X 9HNNU     =I3/ 6X 9HCHORD   =F12.7/ PAR1  075
     6           6X  9HMACH NO.=F12.8/ 6X 9HTCNST   =F12.7/            PAR1  076
     7           6X  9HPSF NO. = A4///)                                PAR1  077
      IF (FLG03.GT.0) WRITE (6,16)                                     PAR1  078
   16 FORMAT (13X 21HSURFACE OF REVOLUTION )                           PAR1  079
      IF (FLG04.GT.0) WRITE (6,20)                                     PAR1  080
   20 FORMAT (13X  9HCROSSFLOW)                                        PAR1  081
      IF (FLG05.GT.0) WRITE (6,24)                                     PAR1  082
   24 FORMAT (13X 15HOFF-BODY POINTS )                                 PAR1  083
      IF (FLG06.GT.0) WRITE (6,28)                                     PAR1  084
   28 FORMAT (13X 15HBASIC DATA ONLY )                                 PAR1  085
      IF (FLG07.GT.0) WRITE (6,32)                                     PAR1  086
   32 FORMAT (13X 17HELLIPSE GENERATOR )                               PAR1  087
      IF (FLG08.GT.0) WRITE (6,36)                                     PAR1  088
   36 FORMAT (13X 14HPRINT MATRICES )                                  PAR1  089
      IF (FLG09.GT.0) WRITE (6,40)                                     PAR1  090
   40 FORMAT (13X 10HOLD SEIDEL )                                      PAR1  091
      IF (FLG10.GT.0) WRITE (6,44)                                     PAR1  092
   44 FORMAT(13X,31HMODIFIED SEIDEL MATRIX SOLUTION)                   PAR1  093
      IF (FLG11.GT.0) WRITE (6,48)                                     PAR1  094
   48 FORMAT (13X 18HPERTURBATIONS ONLY )                              PAR1  095
      IF (FLG12.GT.0) WRITE (6,52)                                     PAR1  096
   52 FORMAT (13X 22HSOLVE POTENTIAL MATRIX )                          PAR1  097
      IF (FLG13.GT.0) WRITE (6,56)                                     PAR1  098
   56 FORMAT (13X 47HMATRIX SOLUTION BY TRIANGULARIZATION   (SOLVIT))  PAR1  099
      IF (FLG14.GT.0) WRITE (6,53)                                     PAR1  100
   53 FORMAT ( 13X 30HPRESCRIBED TANGENTIAL VELOCITY )                 PAR1  101
      IF (FLG18.GT.0) WRITE (6,69)                                     PAR1  102
   69 FORMAT ( 15X 22HWITH SURFACE VORTICITY )                         PAR1  103
      IF (FLG15.GT.0) WRITE (6,54)                                     PAR1  104
   54 FORMAT (13X 12HSTRIP VORTEX )                                    PAR1  105
```

```
      IF (FLG16.GT.0) WRITE (6,64)                                      PAR1  106
   64 FORMAT (13X 40HOMIT AXI-SYMMETRIC UNIFORM FLOW SOLUTION )         PAR1  107
      IF (FLG17.GT.0) WRITE (6,68)                                      PAR1  108
   68 FORMAT (13X 36HOMIT CROSSFLOW UNIFORM FLOW SOLUTION )             PAR1  109
      IF (FLG19.GT.0) WRITE (6,72)                                      PAR1  110
   72 FORMAT (13X 20HPRESCRIBED VORTICITY)                              PAR1  111
      IF (FLG20 .GT. 0)WRITE(6,74)                                      PAR1  112
   74 FORMAT(13X 15HTOTAL VORTICITY  )                                  PAR1  113
      IF (FLG21 .GT. 0)WRITE(6,76)                                      PAR1  114
   76 FORMAT ( 13X  16HEXTRA CROSS FLOW )                               PAR1  115
      IF (FLG22 .GT. 0)WRITE(6,78)                                      PAR1  116
   78 FORMAT(13X 82HGENERATED BOUNDARY CONDITIONS FOR 3 AXISYMMETRIC, 1 PAR1  117
     1CROSS, AND 1 EXTRA CROSS FLOW.  )                                 PAR1  118
      IF (FLG23 .LE. 0)GO TO 81                                         PAR1  119
      WRITE(6,79)                                                       PAR1  120
   79 FORMAT(13X 16HRING WING OPTION )                                  PAR1  121
      FLG03 = 1                                                         PAR1  122
      FLG13 = 1                                                         PAR1  123
      FLG15 = 1                                                         PAR1  124
      FLG19 = 1                                                         PAR1  125
   81 IF (FLG19 .GT. 0)FLG18 = 1                                        PAR1  126
      IF (FLG22.GT.0.AND.NB.NE.2) GO TO 82                              PAR1  127
      GO TO 84                                                          PAR1  128
   82 WRITE(6,83)                                                       PAR1  129
   83 FORMAT (128H0 WHEN GENERATED RESEP BOUNDARY CONDITIONS ARE USED,NU PAR1  130
     1MBER OF BODIES MUST BE EXACTLY TWO.   YOU GOOFED.   EXECUTION TERM PAR1  131
     2INATING.)                                                         PAR1  132
      STOP                                                              PAR1  133
   84 IF (FLG22.GT.0.AND.NNU.GT.0)GO TO 86                              PAR1  134
      GO TO 88                                                          PAR1  135
   86 WRITE (6,87)                                                      PAR1  136
   87 FORMAT (98H0 GENERATED RESEP BOUNDARY CONDITIONS CANNOT HAVE NON-U PAR1  137
     1NIFORM FLOW INPUT.   EXECUTION TERMINATING.)                       PAR1  138
   88 CONTINUE                                                          PAR1  139
       WRITE ( 6,75 )  NIN                                              PAR1  140
      IF (FLG18.LE.0.OR.FLG14.GT.0) GO TO 125                            PAR1  141
   75  FORMAT( 13X,58HINPUT TAPE NO. FOR COORDINATFS AND NON-UNIFORM FLO  PAR1  142
     1W ONLY = ,  I5    )                                                PAR1  143
      WRITF (6,70)                                                       PAR1  144
   70 FORMAT (1H0//63H FLG14 MUST BE USED WITH FLG18 OR FLG19.  EXECUTIO  PAR1  145
     1N TFRMINATED. )                                                    PAR1  146
      STOP                                                               PAR1  147
  125 IF (NNU.LE.0.OR.FLG14.LE.0) GO TO 130                              PAR1  148
      WRITF (6,60)                                                       PAR1  149
   60 FORMAT (1H0// 49H COLUMNS 2 AND 14 OF FLAG CARD ARE BOTH NON-ZERO,  PAR1  150
     A            / 43H ILLEGAL COMBINATION. EXECUTION TERMINATED. )     PAR1  151
      STOP                                                               PAR1  152
C                * READ DATA AND SETUP FOR UNIFORM FLOW                  PAR1  153
  130 CALL BASIC1                                                        PAR1  154
C*** ***NSIGA AND NSIGC ULTIMATELY BECOME THE NUMBER OR RIGHT HAND SIDES  PAR1  155
C*** ***IN AXISYMMFTRIC FLOW AND CROSS FLOW RESPECTIVELY                  PAR1  156
  133 NSIGA=0                                                            PAR1  157
      IF (FLG03.GT.0.AND.FLG16.LE.0) NSIGA=1                             PAR1  158
      NSIGC=0                                                            PAR1  159
      IF (FLG04.GT.0.AND.FLG17.LE.0) NSIGC=1                             PAR1  160
      IF (FLG22.GT.0) GO TO 136                                          PAR1  161
      DO 135 I = 1,5                                                     PAR1  162
      NUNA(I) = 123456                                                   PAR1  163
  135 TYPEA(I) = 100.                                                    PAR1  164
      IF(FLG23 .GT. 0)GO TO 141                                          PAR1  165
      GO TO 138                                                          PAR1  166
C***   ***PREPARE NUNA AND TYPEA FOR NON-UNIFORM AXISYMMETRIC FLOW,GENER  PAR1  167
C***   ***(RESEP) BOUNDARY CONDITIONS                                    PAR1  168
  136 DO 137 I = 1,3                                                     PAR1  169
      NUNA(I) = I                                                        PAR1  170
  137 TYPEA(I) = 100.0                                                   PAR1  171
      GO TO 138                                                          PAR1  172
C                                                                        PAR1  173
C                                                                        PAR1  174
C   ***   RING WING OPTION                                               PAR1  175
```

```
C    ***    STRIP VORTEX FLOWS ALREADY HAVE NUNA(I) = 123456.
C    ***    MAKE PRESCRIBED VORTICITY FLOWS NUNA(J) = TO THEIR FLOW NO.  J
C
  141 ICNT = 0
      DO 142 I = 1,NB
      IF(NLF(I) .GT. 0) GO TO 142
      ICNT = ICNT + 1
  142 CONTINUE
C
C    ***    ICNT IS THE NUMBER OF LIFTING BODIES
C    ***    NUMBER OF FLOWS IS 2 * ICNT  + 1
C
      NFLOWS = 2 * ICNT + 1
      ICNTP2 = ICNT + 2
      DO 143 I = ICNTP2,NFLOWS
  143 NUNA(I) = I
  138 CONTINUE
C*** ***IF FLG02 (NON-UNIFORM FLOW) IS NOT CHECKED INITIALLY, THE FLOW
C*** ***OF CONTROL WILL NEVER REACH BASIC2
      IF (NNU) 140,150,140
C                * READ DATA AND SETUP FOR NON-UNIFORM FLOW
  140 CALL BASIC2
  150 CONTINUE
  160 REWIND 4
      IF (NSIGA .LE. 5 )GO TO 180
  170 WRITE(6,172)
  172 FORMAT (1H1 75HAXI-SYMMETRIC OR CROSSFLOW NON-UNIFORM FLOWS EXCEED
     A 5. EXECUTION TERMINATED  )
      STOP
  180 IF (NSIGC .GT. 5)GO TO 170
      IF (FLG15.LE.0.OR.FLG03.GT.0) GO TO 200
      WRITE (6,190)
  190 FORMAT (64H1STRIP RING VORTEX OPTION MUST USE SURFACE OF REVOLUTIO
     1N OPTION. / 22H EXECUTION TERMINATED. )
      STOP
  200 IF (FLG15.LE.0) GO TO 230                                           PAR1  211
      J = 0                                                               PAR1  212
      DO 210 I = 1, NB                                                    PAR1  213
  210 IF (NLF(I).LE.0) J=J+1                                              PAR1  214
      IF (NSIGA + J  .LE.  5 )GO TO 230                                   PAR1  215
      WRITE (6,220)                                                       PAR1  216
  220 FORMAT (68H1GENERATED STRIP VORTEX ONSET FLOWS (ONE FOR EACH LIFTI  PAR1  217
     1NG BODY) PLUS / 34H INPUT NON-UNIFORM FLOWS EXCEED 5. /             PAR1  218
     2 22H0EXECUTION TERMINATED. )                                        PAR1  219
      STOP                                                                PAR1  220
  230 IF (FLG06.NE.0) GO TO 235                                           PAR1  221
      CALL MATRIX                                                         PAR1  222
C 235 RETURN                                                              PAR1  223I
  235 CONTINUE                                                            PAR1  224C
      END                                                                 PAR1  225
      SUBROUTINE BASIC1
C
C              * READ DATA AND SETUP FOR UNIFORM FLOW
C
       COMMON / NBSAVE / NBOLD, NIN
      COMMON       HEDR(10)  ,CASE       ,NB        ,NNU
     1             ,FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07
     2             ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12
     3             ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17
     4             ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22
     5             ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27
      COMMON   NT,         ND(11),     MN,        NUNA(5),   TYPEA(5),
     1         NER1,       NER2,       NMA,       NSIGA,     NSIGC,
     2         NUNC(5),    TYPFC(5),   NLF(11),   IEC,       NSIGEC,
     3         TYPFEC(5),NUNEC(5)
C     DOUBLE PRECISION  HEDR, CASE
      INTEGER      FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07
     1             ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12
     2             ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17
     3             ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22
     4             ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27
      DIMENSION COSSQR(100), RHS(100)
      REAL         MN
C
      COMMON /CL/   X1(100),   Y1(100),   X2(100),   Y2(100),   DELS(100),
     1              SINA(100),COSA(100),XP(100),   YP(100)
     2             ,XWAKE(11),YWAKE(11)
      COMMON /TL/   TX1(100),  TY1(100),  NG(100),   TG(100),   ALFA(100),
     1  RSDS(100),DALF(100),CHORD,TCNST,DUMMY(1315)
      INTEGER      BDN        ,SUBKS
      REAL         MX         ,MY         ,NG
C
C              * START
      NT=0
      K=0
      K2=NB
       IF( NIN .EQ. 0 )       NIN = 10
      IF (FLG05.NF.0) K2=NB+1
C              * MAJOR LOOP * NO. OF BODIES + OFF BODY POINTS
      LCNT = 0
      DO 1000 L=1,K2
      READ (5,15) NN, MX, MY, THETA, ADDX, ADDY
   15 FORMAT ( 5X I5, 5F10.0)
      READ (5,16) BDN,SUBKS,NLF(L),XF,YE
   16 FORMAT (3(5X,I5),2F10.0)
C*** ***ND(L) IS THE NUMBER OF POINTS ON BODY L, OR THE NUMBER OF OFF
C*** ***BODY POINTS FOR L = NB + 1
      ND(L)=NN
      M=NN-1
      IF (SUBKS) 140,150,140
 140   IF( L .NE. K2 )        GO TO 148
       NTIMES = NBOLD - NB
       IF( NTIMES .LF. 0 )        GO TO 148
       DO 145 NSKIPS = 1, NTIMES
 145   READ(13) ( TX1(I),I=1,NN), (TY1(I),I=1,NN )
 148   READ(13) ( TX1(I),I=1,NN), (TY1(I),I=1,NN )
      GO TO 220
  150 IF (BDN.EQ.0) GO TO 200
      IF (FLG07) 160,200,160
C              * ELLIPSE GENERATOR FOR X1 AND Y1
  160 IF (XE.EQ.0.0) XE=1.
      IF (YF.EQ.0.0) YE=1.
      EN=M
      DGAM=3.141593  /FN
      GAM=3.141593
      DO 170 I=1,NN
      TX1(I)=XE*COS(GAM)
      TY1(I)=YE*SIN(GAM)
  170 GAM=GAM-DGAM
      GO TO 210
```

```
C               * READ X1 AND Y1 FROM INPUT CARDS
  200 DO 204 I=1,NN,6
      READ(NIN,20)TX1(I),TX1(I+1),TX1(I+2),TX1(I+3),TX1(I+4),TX1(I+5)
   20 FORMAT ( 6F10.0)
  204 CONTINUE
      DO 206 I=1,NN,6
      READ(NIN,20)TY1(I),TY1(I+1),TY1(I+2),TY1(I+3),TY1(I+4),TY1(I+5)
  206 CONTINUE
C
C*** *  NB - FLG14 + 1  TO  NB  ARE PRESCRIBED VORTICITY BODIES
C
      IF ( FLG23 .LE. 0  .OR.  (L.LE.NB-FLG14  .OR. L .GT. NB))GO TO 210
C
C*** *  IF CONTROL REACHES THIS POINT, RING WING OPTION IS IN EFFECT AND
C*** *L IS A PRESCRIBED VORTICITY BODY
C   ***   LCNT IS THE RELATIVE NUMBER OF THE WAKE BODY STARTING WITH 1
C
      LCNT = LCNT + 1
      XWAKE(LCNT) = TX1(NN)
      YWAKE(LCNT) = TY1(NN)
C               * SAVE X1 AND Y1 FOR SUBCASE
  210 WRITE (13) (TX1(I),I=1,NN),(TY1(I),I=1,NN)
C               * BASIC DATA CALC. AND PRINT (UNTRANSFORMED COORDINATES)
  220 WRITE (6,24) HEDR, NN, MX, MY, THETA, ADDX, ADDY, XE, YE
   24 FORMAT ( 1H1 25X 26HDOUGLAS  AIRCRAFT  COMPANY /
     1         28X 21HLONG  BEACH  DIVISION /// 5X 10A6 //
     2          8X 4HNN = I4, 15X 4HMX = F13.7, 4X 4HMY = F13.7 /
     3          5X 7HTHETA = F13.7, 4X 6HADDX = F13.7, 2X 6HADDY =F13.7/
     4          8X 4HXE = F13.7, 6X 4HYE = F13.7 )
      IF (BDN) 240,230,240
  230 WRITE (6,28) (I, TX1(I), TY1(I), I=1,NN)
   28 FORMAT ( 1H0 4X 36HOFF-BODY COORDINATES (UNTRANSFORMED) //
     1         10X 5HX-OFF  9X 5HY-OFF // (1H  I3, 2F14.7))
      GO TO 270
  240 SUMS=0.0
```

```
      DO 250 I=1,M
      T1=TX1(I+1)-TX1(I)
      T2=TY1(I+1)-TY1(I)
      X2(I)=(TX1(I+1)+TX1(I))/2.
      Y2(I)=(TY1(I+1)+TY1(I))/2.
      DELS(I)=SQRT(T1*T1+T2*T2)
      SUMS=SUMS+DELS(I)
      RSDS(I)=SUMS
  250 ALFA(I) = ATAN2( T2, T1 )
      MA=M-1
      DO 260 I=1,MA
  260 DALF(I) = ( ALFA(I+1)-ALFA(I) ) * 57.29578
      WRITE (6,36) BDN,TX1(1),TY1(1),X2(1),Y2(1),DELS(1),RSDS(1)
   36 FORMAT ( 1H0 4X 35HON-BODY COORDINATES (UNTRANSFORMED) /
     1         9H BODY NO. I3// 11X 2H X 13X 1HY 11X 7HDELTA S 7X
     2         5HSUMDS 8X 7HD ALPHA //  1H 3H  1,2F14.7 / 4X 4F14.7)
      WRITE (6,40) (I, TX1(I), TY1(I), DALF(I-1), X2(I), Y2(I),
     1           DELS(I), RSDS(I), I=2,M) , NN, TX1(NN), TY1(NN)
   40 FORMAT ( 1H  I3, 2F14.7, 28X F14.7 / 4X 4F14.7)
C              * ADJUST COORDINATES (TRANSFORMED)
  270 IF (MX) 280,300,280
  280 DO 290 I=1,NN
  290 TX1(I)=TX1(I)*MX
  300 IF (MY) 310,330,310
  310 DO 320 I=1,NN
  320 TY1(I)=TY1(I)*MY
  330 IF (THETA) 340,360,340
  340 THETA = THETA / 57.29578
      CSTHT = COS(THETA)
      SNTHT = SIN(THETA)
      DO 350 I=1,NN
      T1=TX1(I)
      TX1(I)=T1*CSTHT+TY1(I)*SNTHT
  350 TY1(I)=TY1(I)*CSTHT-T1*SNTHT
  360 IF (ADDX) 370,390,370
  370 DO 380 I=1,NN
  380 TX1(I)=TX1(I)+ADDX
  390 IF (ADDY) 400,420,400
  400 DO 410 I=1,NN
  410 TY1(I)=TY1(I)+ADDY
  420 IF (CHORD .EQ. 1.0 .OR. CHORD .EQ. 0.0 )GO TO 450
  430 DO 440 I=1,NN
      TX1(I)=TX1(I)/CHORD
  440 TY1(I)=TY1(I)/CHORD
  450 IF (MN) 460,475,460
  460 SRM=SQRT(1.-MN*MN)
      DO 470 I=1,NN
  470 TX1(I)=TX1(I)/SRM
C                * SHIFT X1 AND Y1 TO COMMON /CL/
C*** ***IF BDN = 0.0, OFF BODY POINTS ARE BEING OPERATED ON
  475 IF (BDN) 500,480,500
  480 DO 490 I=1,NN
      XP(I)=TX1(I)
  490 YP(I)=TY1(I)
      WRITE (12) (XP(I),I=1,NN),(YP(I),I=1,NN)
      GO TO 1000
  500 DO 510 I=1,NN
      K=K+1
      X1(K)=TX1(I)
  510 Y1(K)=TY1(I)
      NT=NT+M
 1000 CONTINUE
      REWIND 13
      IF (FLG14.LE.0) GO TO 2000
      IF (FLG14.LE.NB) GO TO 1050
      WRITE (6,1025)
 1025 FORMAT (45H1VALUE OF FLG14 EXCEEDS NO. OF BODIES.  STOP. )
      STOP
 1050 IF (FLG14.NE.NB) GO TO 1075
      NMA=0
      GO TO 1150
 1075 L = NB-FLG14
      NMA = -L
      DO 1100 I = 1, L
C*** ***NMA BECOMES THE NUMBER OF ELEMENTS ON THE 1ST L BODIES (IE THOSE
C*** ***NOT HAVING AN INPUT VORTICITY OR VELOCITY)
 1100 NMA = NMA + ND(I)
C*** ***NR BECOMES THE NUMBER OF ELEMENTS RECEIVING AN INPUT VORTICITY
C*** ***OR VELOCITY
 1150 NR = NT-NMA
      IF (TCNST.GT.0.)GO TO 2000
      DO 1200 I = 1,NR,6
      READ (5,20) TG(I),TG(I+1),TG(I+2),TG(I+3),TG(I+4),TG(I+5)
 1200 CONTINUE
C               * CALC. PARAMETERS WITH TRANSFORMED COORDINATES AND
C                 MACH NO. ADJUSTMENT
 2000 N1=0
      J1=0
      DO 2500 K=1,NB
      M1=N1+1
      N1=N1+ND(K)-1
      DO 2400 J=M1,N1
      J1=J1+1
      T1=X1(J1+1)-X1(J1)
      T2=Y1(J1+1)-Y1(J1)
      X2(J)=(X1(J1+1)+X1(J1))/2.
      Y2(J)=(Y1(J1+1)+Y1(J1))/2.
      DELS(J)=SQRT(T1*T1+T2*T2)
      COSA(J)=T1/DELS(J)
 2400 SINA(J)=T2/DELS(J)
 2500 J1=J1+1
C               * SAVE PARAMETERS
      WRITE (12) (X1(I),I=1,J1),(Y1(I),I=1,J1),(X2(I),I=1,NT)
     1          ,(Y2(I),I=1,NT),(DELS(I),I=1,NT)
      REWIND 12
```

```
C                * SAVE SINA AND COSA ON TAPE 4 FOR CALC. OF MATRIX
C                  SOLUTION (RIGHT HAND MATRIX)
      WRITF (4)  (SINA(I),I=1,NT),(COSA(I),I=1,NT)
      IF ( FLG14) 2600,2600,2550
 2550 IF (TCNST.GT.0.0) WRITE(4) (TCNST,I=1,NR)
      IF (TCNST.LE.0.) WRITE(4) (TG(I), I=1,NR)
 2600 IF (FLG22.LE.0) RETURN
      NPB1 = ND(1) - 1
      DO 2700 I = 1,NPB1
      COSSQR(I) = COSA(I)**2
 2700 RHS(I) = 2.0 * ABS( SINA(I) * COSA(I) )
      WRITE(4) ( COSSQR(I),I=1,NPB1), (RHS(I),I = 1,NPB1)
      RETURN
      END
      SUBROUTINE BASIC2                                                 BAS2  001
C                                                                       BAS2  002
C                * READ DATA AND SETUP FOR NON-UNIFORM FLOWS            BAS2  003
C                                                                       BAS2  004
       COMMON / NBSAVE / NBOLD, NIN                                     BAS2  005
      COMMON       HEDR(10)   ,CASE       ,NB        ,NNU               BAS2  006
     1            ,FLG03      ,FLG04      ,FLG05     ,FLG06     ,FLG07  BAS2  007
     2            ,FLG08      ,FLG09      ,FLG10     ,FLG11     ,FLG12  BAS2  008
     3            ,FLG13      ,FLG14      ,FLG15     ,FLG16     ,FLG17  BAS2  009
     4            ,FLG18      ,FLG19      ,FLG20     ,FLG21     ,FLG22  BAS2  010
     5            ,FLG23      ,FLG24      ,FLG25     ,FLG26     ,FLG27  BAS2  011
      COMMON   NT,        ND(11),    MN,        NUNA(5),  TYPEA(5),     BAS2  012
     1         NER1,      NER2,      NMA,       NSIGA,    NSIGC,        BAS2  013
     2         NUNC(5),   TYPEC(5),  NLF(11),   IEC,      NSIGEC,       BAS2  014
     3         TYPFEC(5),NUNEC(5)                                       BAS2  015
C     DOUBLE PRECISION  HEDR, CASE                                      BAS2  016I
      INTEGER      FLG03      ,FLG04      ,FLG05     ,FLG06     ,FLG07  BAS2  017
     1            ,FLG08      ,FLG09      ,FLG10     ,FLG11     ,FLG12  BAS2  018
     2            ,FLG13      ,FLG14      ,FLG15     ,FLG16     ,FLG17  BAS2  019
     3            ,FLG18      ,FLG19      ,FLG20     ,FLG21     ,FLG22  BAS2  020
     4            ,FLG23      ,FLG24      ,FLG25     ,FLG26     ,FLG27  BAS2  021
      REAL         MN                                                   BAS2  022
C                                                                       BAS2  023
      COMMON /CL/    X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100), BAS2  024
     1               SINA(100),COSA(100),XP(100),  YP(100)              BAS2  025
     2              ,XWAKE(11),YWAKE(11)                                BAS2  026
      COMMON /TL/    TX1(100), TY1(100), NG(100),  TG(100),  ALFA(100), BAS2  027
     1  RSDS(100),DALF(100),CHORD,TCNST,DUMMY(1315)                     BAS2  028
      INTEGER      BDN                                                  BAS2  029
      REAL         MX         ,MY         ,NG                           BAS2  030
C                                                                       BAS2  031
C                * START                                                BAS2  032
C                * SETS OF NON-UNIFORM FLOW LOOP                        BAS2  033
      NSIGEC = 0                                                        BAS2  034
      KA=0                                                              BAS2  035
      KC=0
      KEC = 0
      DO 1000 L=1,NNU
      READ (5,20) NUN,MSF,TYPE,FG
   20 FORMAT ( 2(5X I5), 2F10.0)
      IF (MSF.EQ.1.OR.MSF.EQ.2.OR.MSF.EQ.5) GO TO 30
      KA=KA+1
      NSIGA=NSIGA+1
      NUNA(KA)=NUN
      TYPEA(KA)=TYPE
   30 IF (MSF.EQ.0.OR.MSF.EQ.2.OR.MSF.EQ.4) GO TO 35
      KC=KC+1
      NSIGC=NSIGC+1
      NUNC(KC)=NUN
      TYPEC(KC)=TYPE
   35 IF (MSF.LT.2.OR.MSF.EQ.3) GO TO 40
      KEC = KEC + 1
      NSIGEC = NSIGEC + 1
      NUNEC(KEC) = NUN
      TYPEEC(KEC) = TYPE
   40 IF (TYPE) 50,70,70
C                * COMPUTED TYPE
   50 DO 60 I=1,NT
      NG(I)=Y2(I)
   60 TG(I)=FG-X2(I)
      GO TO 110
C                * (X,Y) OR (N,T) TYPE * READ INPUT
   70 DO 90 I=1,NT,6
      READ( 5 ,80)NG(I),NG(I+1),NG(I+2),NG(I+3),NG(I+4),NG(I+5)
   80 FORMAT ( 6F10.0)
   90 CONTINUE
      DO 100 I=1,NT,6
      READ( 5 ,80)TG(I),TG(I+1),TG(I+2),TG(I+3),TG(I+4),TG(I+5)
  100 CONTINUE
  110 IF (TYPE) 120,140,120
  120 DO 130 I = 1, NT
      T1=NG(I)
      NG(I)= T1*SINA(I)-TG(I)*COSA(I)
  130 TG(I)= T1*COSA(I)+TG(I)*SINA(I)
C               * WRITE BASIC DATA OUTPUT
  140 WRITE (6,150) HEDR,MSF,TYPE,FG,NUN,(NG(I),I=1,NT)
  150 FORMAT ( 1H1 25X 26HDOUGLAS  AIRCRAFT  COMPANY /
     1         28X, 21HLONG  BEACH  DIVISION /// 5X 10A6 //
     2         6X 5HMSF = I4, 10X 6HTYPE = F10.4, 10X 4HFG = F13.7 /
     3          1H0, 4X, 20HNON-UNIFORM FLOW NO.I6 /
     4          1H0, 4X, 10HLIST OF NG// (1H 6F14.7))
      WRITE (6,160) (TG(I), I = 1, NT)
  160 FORMAT (1H0 4X 10HLIST OF TG // (1H 6F14.7))
      WRITE (4) MSF,(NG(I),I=1,NT),(TG(I),I=1,NT)
 1000 CONTINUE
      RETURN
      END
      SUBROUTINE MATRIX                                                 MATX  001
C                                                                       MATX  002
C               * COMPUTE MATRIX A,B,Z OR X,Y,Z                         MATX  003
C                                                                       MATX  004
      COMMON       HEDR(10)   ,CASE       ,NB         ,NNU              MATX  005
     1            ,FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07MATX  006
     2            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12MATX  007
     3            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17MATX  008
     4            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22MATX  009
     5            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27MATX  010
      COMMON    NT,         ND(11),    MN,         NUNA(5),   TYPEA(5), MATX  011
     1          NER1,       NER2,      NMA,        NSIGA,     NSIGC,    MATX  012
     2          NUNC(5),    TYPEC(5),  NLF(11),    IEC,       NSIGEC,   MATX  013
     3          TYPEEC(5),NUNEC(5)                                      MATX  014
C     DOUBLE PRECISION  HEDR, CASE                                      MATX  015I
      INTEGER      FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07MATX  016
     1            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12MATX  017
     2            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17MATX  018
     3            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22MATX  019
     4            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27MATX  020
      REAL         MN                                                   MATX  021
      LOGICAL   PF                                                      MATX  022
C                                                                       MATX  023
      COMMON /ECF/ ECX(100), ECY(100), ECZ(100)                         MATX  024
      COMMON  /RNGWNG/ VA(100,2),  VR(100,2),VAN(100), VAT(100)         MATX  025
      COMMON /CL/    X1(100),   Y1(100),   X2(100),   Y2(100),   DELS(100),MATX  026
     1               SINA(100),COSA(100),XP(100),   YP(100)             MATX  027
     2              ,XWAKE(11),YWAKE(11)                                MATX  028
      COMMON /TL/    A(100),    B(100),    AX(100),   AY(100),   AZ(100),MATX  029
     1               CX(100),   CY(100),   CZ(100),   AXV(100),AYV(100),MATX  030
     2               VN(100,5),VT(100,5),BON,                  IAC,     MATX  031
     3               I,         J,         J1,         SJ,        DS,   MATX  032
     4               DX,        DY,        NI,         XJ,        YJ,   MATX  033
     5               XK,        EEK,       EKK,        K,         PF    MATX  034
C                                                                       MATX  035
```

```
C                  * START                                                MATX  036
C                  * INITIALIZE                                           MATX  037
      L1=NT                                                               MATX  038
      BON=0.0                                                             MATX  039
      YZERO=0.0                                                           MATX  040
C***   ***TEST TYPE OF FLOW AND SET INDICATORS IAC AND IEC                 MATX  041
C***   ***CROSS FLOW ONLY                          IAC = -1  IEC = -1     MATX  042
C***   ***AXISYMMETRIC FLOW ONLY                   IAC = +1  IEC = -1     MATX  043
C***   ***EXTRA CROSS FLOW ONLY                    IAC =  0  IEC =  0     MATX  044
C***   ***CROSS FLOW AND AXISYMMETRIC FLOW         IAC =  0  IEC = -1     MATX  045
C***   ***CROSS FLOW AND EXTRA CROSS FLOWW         IAC = -1  IEC = +1     MATX  046
C***   ***AXISYMMETRIC AND EXTRA CROSS FLOW        IAC = +1  IEC = +1     MATX  047
C***   ***AXISYMMETRIC, CROSS, AND EXTRA CROSS     IAC=0 IEC = +1         MATX  048
      IF(FLG03)30,10,30                                                   MATX  049
   10 IF(FLG04)25,15,25                                                   MATX  050
   15 IAC = 0                                                             MATX  051
      IEC = 0                                                             MATX  052
      GO TO 55                                                            MATX  053
   25 IAC = -1                                                            MATX  054
      GO TO 45                                                            MATX  055
   30 IF(FLG04)35,40,35                                                   MATX  056
   35 IAC = 0                                                             MATX  057
      GO TO 45                                                            MATX  058
   40 IAC = 1                                                             MATX  059
   45 IF(FLG21)50,53,50                                                   MATX  060
   50 IEC = +1                                                            MATX  061
      GO TO 55                                                            MATX  062
   53 IEC = -1                                                            MATX  063
   55 ASSIGN 110 TO K1                                                    MATX  064
      IF (FLG15.GT.0) ASSIGN 102 TO K1                                    MATX  065
   60 DO 70 I=1,L1                                                        MATX  066
      DO 65 J = 1,5                                                       MATX  067
      VN(I,J) = 0.                                                        MATX  068
   65 VT(I,J) = 0.                                                        MATX  069
      VAN(I)=0.0                                                          MATX  070
   70 VAT(I)=0.0
C               *  I MIDPOINT LOOP
      DO 400 I=1,L1
C               *  J ELEMENT LOOP
C                  J1 IS THE COORDINATE COUNTER
C                  J IS THE ELEMENT COUNTER
      J1=0
      N1=0
      IF (FLG23 .GT. 0)CALL NOTS
      DO 110 K=1,NB
      M1=N1+1
      N1=N1+ND(K)-1
      DO 100 J=M1,N1
      J1=J1+1
      PF = FLG18.GT.0.AND.J.GT.NMA.OR.FLG20.GT.0
C               * COMPUTE X,Y,Z MATRICES
      CALL XYZ
  100 CONTINUE
      GO TO K1, (102,110)
  102 IF (NLF(K).GT.0) GO TO 110
      IF (BON.EQ.0.) GO TO 105
      DO 103 J = M1, N1
      VN(I,K) = VN(I,K)+AXV(J)
  103 VT(I,K) = VT(I,K)+AYV(J)
      GO TO 110
  105 DO 106 J = M1, N1
      VN(I,K) = VN(I,K) +AXV(J)*SINA(I) - AYV(J)*COSA(I)
  106 VT(I,K) = VT(I,K) +AXV(J)*COSA(I) + AYV(J)*SINA(I)
  110 J1=J1+1
      IF( FLG08 .LE.0  .OR.  FLG15 .LE. 0 )GO TO 118
C
C*** *  PRINT STRIP VORTEX MATRICES
C
      IF( I .EQ. 1  .AND.  BON .EQ. 0. )WRITE(6,111)
      IF( I .EQ. 1  .AND.  BON .EQ. 1. )WRITE(6,112)
  111 FORMAT(1H1,31H STRIP VORTEX MATRICES  ON BODY //)                MATX  106
  112 FORMAT(1H1,31H STRIP VORTEX MATRICES OFF BODY //)                MATX  107
      WRITF(6,114)I,( AXV(J),J=1,NT)                                   MATX  108
      WRITE(6,115)  ( AYV(J),J=1,NT)                                   MATX  109
  114 FORMAT(1H0,5H ROW,I4/9H X MATRIX / (6E20.7) )                    MATX  110
  115 FORMAT(9H Y MATRIX / (6E20.7) )                                  MATX  111
  118 IF (BON)120,210,120                                              MATX  112
C                * SAVE X,Y,Z ON TAPE *OFF BODY POINTS                 MATX  113
C***    ***SAVE X,Y,Z ON TAPE    *    OFF BODY POINTS                  MATX  114
C***    ***AXISYMMETRIC FLOW    *    TAPE 9                           MATX  115
C***    ***CROSS FLOW    *    TAPE 10                                  MATX  116
C***    ***EXTRA CROSS FLOW    *    TAPE 8                             MATX  117
  120 IF(IEC.EQ.-1)GO TO 125                                           MATX  118
  122 WRITF(8) (ECX(J),J=1,NT), (ECY(J),J=1,NT), (FCZ(J),J=1,NT)       MATX  119
      IF (IEC) 125,400,125                                             MATX  120
  125 IF(IAC) 140,130,130                                              MATX  121
  130 WRITE (9) (AX(J),J=1,NT),(AY(J),J=1,NT),(AZ(J),J=1,NT)           MATX  122
      IF (IAC) 400,140,400                                             MATX  123
  140 WRITE (10)(CX(J),J=1,NT),(CY(J),J=1,NT),(CZ(J),J=1,NT)           MATX  124
      GO TO 400                                                        MATX  125
C***    ***SAVE ON TAPE    *    ON BODY                                MATX  126
C***    ***AXISYMMETRIC FLOW    *    TAPE 9                           MATX  127
C***    ***CROSS FLOW    *    TAPE 10                                  MATX  128
C***    ***EXTRA CROSS FLOW    *    TAPE 8                             MATX  129
C***    ***IEC = -1 MEANS NO EXTRA CROSS FLOW                          MATX  130
  210 IF (IEC.EQ.-1) GO TO 240                                         MATX  131
  220 DO 230 J = 1,NT                                                  MATX  132
      A(J) = -ECX(J) * SINA(I) + ECY(J) * COSA(I)                      MATX  133
  230 B(J) = FCX(J) * COSA(I) + ECY(J) * SINA(I)                       MATX  134
      WRITE (8) (A(J),J=1,NT), (B(J),J=1,NT), (ECZ(J),J=1,NT)          MATX  135
      IF ( IEC ) 240,400,240                                           MATX  136
  240 IF (IAC) 310,250,250                                             MATX  137
  250 DO 260 J=1,NT                                                    MATX  138
      A(J)=-AX(J)*SINA(I)+AY(J)*COSA(I)                                MATX  139
  260 B(J)=AX(J)*COSA(I)+AY(J)*SINA(I)                                 MATX  140
```

```
      WRITE (9) (A(J),J=1,NT),(B(J),J=1,NT),(AZ(J),J=1,NT)
  270 IF (IAC) 400,310,400
  310 DO 320 J=1,NT
      A(J)=-CX(J)*SINA(I)+CY(J)*COSA(I)
  320 B(J)=CX(J)*COSA(I)+CY(J)*SINA(I)
      WRITE (10) (A(J),J=1,NT),(B(J),J=1,NT),(CZ(J),J=1,NT)
  400 CONTINUE
      IF (FLG15.LE.0) GO TO 1400
      IF (BON.NE.0.) GO TO 1200
C*** ***ON BODY
      READ (4)
C
C*** *  IF FLG23 .GT. 0 INPUT NNU MUST BE NONE, HENCE NNU = 0 HERE
C
      IF (NNU.LE.0) GO TO 600
      DO 500 I = 1, NNU
      READ (4) MSF,(A(J),J=1,NT),(B(J),J=1,NT)
  500 WRITE (3) MSF,(A(J),J=1,NT),(B(J),J=1,NT)
      REWIND 3
      REWIND 4
      READ (4)
  600 N=NSIGA-1
      IF (FLG16.GT.1) N=NSIGA
C*** ***N = 0 MEANS 1 RHS ONLY  NO NON-UNIFORM FLOW
C*** *  IF FLG23 .GT. 0 INPUT NNU MUST BE NONE, HENCE N = 0  HERE
C
      IF (N.EQ.0) GO TO 800
      DO 700 I = 1, N
      READ (3) MSF,(A(J),J=1,NT),(B(J),J=1,NT)
  700 WRITE(4) MSF,(A(J),J=1,NT),(B(J),J=1,NT)
  800 M=0
C*** *  SKIP PRESCRIBED VORTEX INPUTS ON 4 SO THAT STRIP VORTEX
C*** *  SUMMATIONS CAN GO BEHIND IT
C
      IF(FLG23 .GT. 0)READ(4)
      DO 900 J = 1, NB
      IF (NLF(J).GT.0) GO TO 900
      NSIGA=NSIGA+1
      NNU=NNU+1
      WRITE (4) M,(VN(I,J),I=1,NT),(VT(I,J),I=1,NT)
  900 CONTINUE
C
C*** *  SINCE NO NNU IS INPUT WITH FLG23 GT 0, NSIGC IS MAX OF 1 AND M
C*** *  SHOLD BE 0 IF FLG17 LE 0.  DONT USE FLG17 WITH FLG23
C
      IF (FLG23 .LE. 0)GO TO 975
C
C*** *  RING WING OPTION - FORM COLUMN (PARTLY) FOR PRESCRIBED VORTICIY
C*** *RHS
C
      IBOD = 0
      DO 950 J=1,NB
      IF( NLF(J) .GT. 0)GO TO 950
      IBOD = IBOD + 1
C
C*** *  CONVERT (ON BODY) X,Y  TO NORMAL, TANGENTIAL
C
      DO 925 I=1,NT
      VAN(I) = VAN(I) + VA(I,IBOD)*SINA(I)  - VR(I,IBOD)*COSA(I)
  925 VAT(I) = VAT(I) + VA(I,IBOD)*COSA(I)  + VR(I,IBOD)*SINA(I)
      WRITE(4)(VAN(I),I=1,NT),(VAT(I),I=1,NT)
  950 CONTINUE
  975 M = NSIGC - 1
      IF (FLG17.GT.0) M=NSIGC
      IF (M.LE.0) GO TO 1100
      DO 1000 I = 1, M
      READ (3) MSF,(A(J),J=1,NT),(B(J),J=1,NT)
 1000 WRITE (4) MSF,(A(J),J=1,NT),(B(J),J=1,NT)
 1100 REWIND 3
      GO TO 1400
```

```
C*** ***OFF BODY
 1200 DO 1300 J = 1, NB
      IF (NLF(J).GT.0) GO TO 1300
      WRITE(4)  (VN(I,J), I = 1,L1), (VT(I,J),I = 1,L1)
 1300 CONTINUE
      IF(FLG23 .LE. 0)GO TO 1400
      IBOD = 0
      DO 1350 J=1,NB
      IF( NLF(J) .GT. 0)GO TO 1350
      IBOD = IBOD + 1
      WRITE(4) ( VA(I,IBOD),I=1,L1 ), ( VR(I,IBOD),I=1,L1)
 1350 CONTINUE
C              * TEST IF OFF BODY COMPLETED
C              * TEST IF OFF BODY
 1400 IF (FLG05.EQ.0.OR.BON.NE.0.) GO TO 1600
C              * INITIAL FOR OFF BODY * THEN RE-ENTER I,J LOOPS
      BON=1.
      L1=ND(NB+1)
      DO 1500 I = 1, L1
      X2(I) = XP(I)
 1500 Y2(I) = YP(I)
      GO TO 60
 1600 REWIND 9
      REWIND 8
      REWIND 10
      REWIND 4
      RETURN
      END
      SUBROUTINE XYZ                                                    XYZ  001
C                                                                       XYZ  002
C              * CONTROL FOR X,Y,Z MATRICES COMPUTATION                 XYZ  003
C                                                                       XYZ  004
      COMMON  /D/ D1, D3, XMXJ, YMYJ, XMXJP1, YMYJP1, S                  XYZ  005
      COMMON       HEDR(10)  ,CASE       ,NU         ,NNU               XYZ  006
     1            ,FLG03     ,FLG04      ,FLG05      ,FLG06     ,FLG07   XYZ  007
     2            ,FLG08     ,FLG09      ,FLG10      ,FLG11     ,FLG12   XYZ  008
     3            ,FLG13     ,FLG14      ,FLG15      ,FLG16     ,FLG17   XYZ  009
     4            ,FLG18     ,FLG19      ,FLG20      ,FLG21     ,FLG22   XYZ  010
     5            ,FLG23     ,FLG24      ,FLG25      ,FLG26     ,FLG27   XYZ  011
      COMMON    NT,        ND(11),     MN,        NUNA(5),   TYPEA(5),  XYZ  012
     1          NER1,      NER2,       NMA,       NSIGA,     NSIGC,     XYZ  013
     2          NUNC(5),   TYPEC(5),   NLF(11),   IEC,       NSIGEC,    XYZ  014
     3          TYPFEC(5),NUNEC(5)                                      XYZ  015
C     DOUBLE PRECISION  HEDR, CASE                                      XYZ  016
      INTEGER      FLG03     ,FLG04      ,FLG05      ,FLG06     ,FLG07   XYZ  017
     1            ,FLG08     ,FLG09      ,FLG10      ,FLG11     ,FLG12   XYZ  018
     2            ,FLG13     ,FLG14      ,FLG15      ,FLG16     ,FLG17   XYZ  019
     3            ,FLG18     ,FLG19      ,FLG20      ,FLG21     ,FLG22   XYZ  020
     4            ,FLG23     ,FLG24      ,FLG25      ,FLG26     ,FLG27   XYZ  021
      REAL         MN                                                   XYZ  022
      LOGICAL PF                                                        XYZ  023
C                                                                       XYZ  024
      COMMON  /RNGWNG/ VA(100,2),  VR(100,2),VAN(100), VAT(100)          XYZ  025
      COMMON /CL/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),  XYZ  026
     1              SINA(100),COSA(100),XP(100),  YP(100)               XYZ  027
     2             ,XWAKE(11),YWAKE(11)                                  XYZ  028
      COMMON /TL/   A(100),   B(100),   AX(100),  AY(100),  AZ(100),    XYZ  029
     1              CX(100),  CY(100),  CZ(100),  AXV(100),AYV(100),    XYZ  030
     2              VN(100,5),VT(100,5),BON,           IAC,             XYZ  031
     3              I,        J,        J1,       SJ,       DS,         XYZ  032
     4              DX,       DY,       NI,       XJ,       YJ,         XYZ  033
     5              XK,       EEK,      EKK,      K,        PF          XYZ  034
C                                                                       XYZ  035
```

```
C                * START                                                   XYZ   036
      IF (BON) 100,10,100                                                 XYZ   037
   10 IF (J-I) 110,20,110                                                 XYZ   038
C                * J EQUAL I PATH                                          XYZ   039
   20 T1=.5*DELS(J)                                                       XYZ   040
      SJ=T1/Y2(J)                                                         XYZ   041
      IF (SJ-.08) 30,30,40                                                XYZ   042
   30 CALL XYZ1                                                           XYZ   043
      GO TO 1000                                                          XYZ   044
   40 SJ=.08                                                              XYZ   045
      CALL XYZ1                                                           XYZ   046
      NI=33                                                               XYZ   047
      T2=.08*Y2(J)                                                        XYZ   048
      DS=(T1-T2)/32.                                                      XYZ   049
      DX=DS*COSA(J)                                                       XYZ   050
      DY=DS*SINA(J)                                                       XYZ   051
      XJ=X2(J)+T2*COSA(J)-DX                                              XYZ   052
      YJ=Y2(J)+T2*SINA(J)-DY                                              XYZ   053
      CALL XYZ2                                                           XYZ   054
      GO TO 300                                                           XYZ   055
C                * INITIAL Y COORDINATE MID-POINT FOR ZERO TEST            XYZ   056
  100 YZERO=Y2(I)-.000001                                                 XYZ   057
C                * J NOT EQUAL I PATH                                      XYZ   058
C                * COMPUTE MINIMUM DISTANCE TO I MIDPOINT                  XYZ   059
  110 J1P1 = J1 + 1                                                       XYZ   060
      XMXJ = X2(I) - X1(J1)                                               XYZ   061
      YMYJ = Y2(I) - Y1(J1)                                               XYZ   062
      XMXJP1 = X2(I) - X1(J1P1)                                           XYZ   063
      YMYJP1 = Y2(I) - Y1(J1P1)                                           XYZ   064
      D1 = XMXJ**2 + YMYJ**2                                              XYZ   065
      D2=(X2(I)-X2(J))**2+(Y2(I)-Y2(J))**2                                XYZ   066
      D3 = XMXJP1**2 + YMYJP1**2                                          XYZ   067
      S = SQRT( (X1(J1P1) - X1(J1) )**2  +   ( Y1(J1P1) - Y1(J1) )**2 )   XYZ   068
      IF (D1-D2) 130,130,120                                              XYZ   069
  120 IF (D2-D3) 150,150,140                                              XYZ   070
  130 IF (D1-D3) 160,160,140
  140 DM=SQRT(D3)
      GO TO 170
  150 DM=SQRT(D2)
      GO TO 170
  160 DM=SQRT(D1)
C                  * COMPUTE NO. OF INTERVALS(NI) AND DELTA S (DS)
C                    FOR SIMPSON RULE INTEGRATION
  170 IF (DM.EQ.0.0) GO TO 200
      NI=8.*DELS(J)/DM+0.9
      IF (NI) 180,180,190
  180 NI=3
      DS=DELS(J)/2.
      GO TO 220
  190 NI=NI+NI
      IF (NI-128) 210,200,200
  200 NI=129
      DS=DELS(J)/128.
      GO TO 220
  210 XNI=NI
      DS=DELS(J)/XNI
      NI=NI+1
  220 DX=DS*COSA(J)
      DY=DS*SINA(J)
  300 XJ=X1(J1)-DX
      YJ=Y1(J1)-DY
      CALL XYZ2
 1000 RETURN
      END
      SUBROUTINE XYZ1
C
C              * COMPUTE X,Y,Z MATRICES FOR SJ LESS THAN OR EQUAL .08
C
      COMMON       HEDR(10)   ,CASE       ,NB         ,NNU
     1            ,FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07
     2            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12
     3            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17
     4            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22
     5            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27
      COMMON    NT,        ND(11),    MN,        NUNA(5),   TYPEA(5),
     1          NER1,      NER2,      NMA,       NSIGA,     NSIGC,
     2          NUNC(5),   TYPEC(5),  NLF(11),   IEC,       NSIGEC,
     3          TYPEEC(5),NUNEC(5)
C     DOUBLE PRECISION  HEDR, CASE
      INTEGER       FLG03     ,FLG04      ,FLG05      ,FLG06      ,FLG07
     1            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12
     2            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17
     3            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22
     4            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27
      COMMON   /RNGWNG/ VA(100,2),  VR(100,2),VAN(100), VAT(100)
      COMMON /ECF/ ECX(100), ECY(100), ECZ(100)
      REAL          MN
      LOGICAL PF
C
      COMMON /CL/    X1(100),   Y1(100),   X2(100),   Y2(100),   DELS(100),
     1               SINA(100),COSA(100),XP(100),   YP(100)
     2              ,XWAKE(11),YWAKE(11)
      COMMON /TL/    A(100),    B(100),    AX(100),   AY(100),   AZ(100),
     1               CX(100),   CY(100),   CZ(100),   AXV(100),AYV(100),
     2               VN(100,5),VT(100,5),BON,             IAC,
     3               I,         J,         J1,        SJ,        DS,
     4               DX,        DY,        NI,        XJ,        YJ,
     5               XK,        FEK,       EKK,       K,         PF
C
```

```
C                 * START
C                 * INITIALIZE
      T1=SJ*SJ
      T2=ALOG(SJ/8.)
      T3=SINA(J)*SINA(J)
      T4=T2+T3
      T5=.6666667  *T3
      T6=T5*T3
      T7=SJ+SJ
      T8=T7+T7
       T9=6.283185 *COSA(J)
      T10=6.283185 *SINA(J)
      T11=T1*SJ
      T14 = .3333333 * (16.0 + 6.0 * T3) + 2.0 * T2
      IF (IEC.EQ.-1) GO TO 15
C***   ***EXTRA CROSS FLOW   1ST TERM OF X(I,I), Y(I,I), Z(I,I)
   10 ECX(J) = 6.283185 * SINA(J) + 2.0 * SINA(J) * COSA(J) * SJ
      ECY(J) =-6.283185 * COSA(J) + SJ * T14
      ECZ(J) = -8.0 * (1.666667 + T2) * SJ
      IF (IEC) 15,1000,15
   15 IF(PF) GO TO 25
      IF (IAC) 30,20,20
C                 * AXIS FLOW
   20 AX(J)=T10+SINA(J)*COSA(J)*(T7+(T4+2.166667  )*T11/12.)
      AY(J)=T7*T4-T9-(1.+T2-T3-T6)*T11/8.
      T12=T1+T1
      AZ(J)=Y2(J)*T8*(1.-T2+T1*(2.-T12+3.*T2*(1.+T12))/144.)
   25 IF (IAC) 30,30,100
C                 * CROSS FLOW
   30 T13=T1/16.
      CX(J)=T10+2.*SINA(J)*SJ*COSA(J)*(1.-T13*(3.-T5+T2+T2))
      CY(J)=-T9+T7*(2.+T4+T13*(1.-4.777778  *T3+T6+T2*(3.-2.666667  *
     1             T3)))
      CZ(J)=-T8*(1.+T2-T13*(1.111111  *T3+T2*(T5-1.)))
  100 IF (PF) GO TO 200
```

```
      IF (FLG15.LE.0.OR.NLF(K).GT.0) GO TO 1000                           XYZ1  071
  200 AXV(J) = T9+T7*(T2-T3)+T11*(T2*(12.*T3-9.)                          XYZ1  072
     1          -9. + 23.*T3 - 6.*T3*T3) / 72.                            XYZ1  073
      AYV(J) = T10 + 2.*COSA(J)*SINA(J)*(SJ-T11*(6.*T2+9.-2.*T3)/48. )    XYZ1  074
      IF (.NOT.PF) GO TO 1000                                             XYZ1  075
      AX(J) = AXV(J)                                                      XYZ1  076
      AY(J) = AYV(J)                                                      XYZ1  077
C                                                                         XYZ1  078
C*** *  RING WING OPTION PV EFFECTS ON ITSELF NEGLECTS 2*PI TERMS          XYZ1  079
C                                                                         XYZ1  080
      IF(FLG23 .LE. 0)GO TO 1000                                          XYZ1  081
      AX(J) = AX(J) - T9                                                  XYZ1  082
      AY(J) = AY(J) - T10                                                 XYZ1  083
      AYV(J)=AY(J)                                                        XYZ1  084
      AXV(J)=AX(J)                                                        XYZ1  085
 1000 CONTINUE                                                            XYZ1  086
      RETURN                                                              XYZ1  087
      END                                                                 XYZ1  088
      SUBROUTINE XYZ2
C
C              * COMPUTE X,Y,Z MATRICES USING SIMPSON RULE INTEGRATION
C
      REAL LIJ2D
      COMMON /D/ R1SQR, R2SQR, XMXJ, YMYJ, XMXJP1, YMYJP1, S
      COMMON      HEDR(10)  ,CASE       ,NB         ,NNU
     1            ,FLG03     ,FLG04      ,FLG05      ,FLG06      ,FLG07
     2            ,FLG08     ,FLG09      ,FLG10      ,FLG11      ,FLG12
     3            ,FLG13     ,FLG14      ,FLG15      ,FLG16      ,FLG17
     4            ,FLG18     ,FLG19      ,FLG20      ,FLG21      ,FLG22
     5            ,FLG23     ,FLG24      ,FLG25      ,FLG26      ,FLG27
      COMMON   NT,       ND(11),    MN,       NUNA(5),  TYPEA(5),
     1         NER1,     NER2,      NMA,      NSIGA,    NSIGC,
     2         NUNC(5),  TYPEC(5),  NLF(11),  IEC,      NSIGEC,
     3         TYPEEC(5),NUNEC(5)
C     DOUBLE PRECISION  HEDR, CASE
      INTEGER     FLG03     ,FLG04      ,FLG05      ,FLG06      ,FLG07
     1            ,FLG08     ,FLG09      ,FLG10      ,FLG11      ,FLG12
     2            ,FLG13     ,FLG14      ,FLG15      ,FLG16      ,FLG17
     3            ,FLG18     ,FLG19      ,FLG20      ,FLG21      ,FLG22
     4            ,FLG23     ,FLG24      ,FLG25      ,FLG26      ,FLG27
      COMMON /ECF/ ECX(100), ECY(100), ECZ(100)
      COMMON  /RNGWNG/ VA(100,2),  VR(100,2),VAN(100), VAT(100)
      DATA NSW /1/
C***   ***RSMALL WILL BE TRUE IF IS .LT. EPS      AND THEREFORE SMALL EL
      LOGICAL RSMALL
      REAL        MN
      LOGICAL PF
C
      COMMON /CL/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),
     1              SINA(100),COSA(100),XP(100),  YP(100)
     2             ,XWAKE(11),YWAKE(11)
      COMMON /TL/   A(100),   B(100),   AX(100),  AY(100),  AZ(100),
     1              CX(100),  CY(100),  CZ(100),  AXV(100),AYV(100),
     2              VN(100,5),VT(100,5),BON,          IAC,
     3              I,        J,        J1,       SJ,       DS,
     4              DX,       DY,       NI,       XJ,       YJ,
     5              XK,       EEK,      EKK,      K,        PF
C
C                   * START
C                   * INITIALIZE
      EPS = 0.0
      ASSIGN 570 TO K1
C*** ***K5 = 80 FOR NON-SMALL ELEMENT AXISYMMETRIC
      ASSIGN 80 TO K5
C***   ***K6 = 295 FOR NON SMALL ELEMENT   CROSS FLOW
      ASSIGN 295 TO K6
      IF (FLG15.LE.0.OR.NLF(K).GT.0) GO TO 15
   10 ASSIGN 420 TO K1
   15 S2=.6666667  *DS
      S1 = .3333333 * DS
      S3 = 8.0/3.0 * S1
      S5 = .3333333 * S1
      S4 = S2+S2
      T1=Y2(I)*Y2(I)
      ASSIGN 28 TO K2
      ASSIGN 410 TO K3
      ASSIGN 570 TO K4
      IF( .NOT. PF  ) GO TO 12
      ASSIGN 110 TO K2
      ASSIGN 420 TO K3
      ASSIGN 560 TO K4
   12 IF ( (I .NE. J) .OR. (BON .NE. 0.0) ) GO TO 16
C*** ***I = J ** ON BODY
      R = DELS(I) / 2.0
      RSMALL = ( R / Y2(I) ) .LT. EPS
      NSW = 2
      GO TO 17
   16 R = SQRT( AMAX1(R1SQR,R2SQR) )
```

```
      IF( ABS(Y2(I) ) .LT. 10E-30)GO TO 13
      RSMALL = ( R / Y2(I) ) .LT. EPS
      GO TO 17
   13 RSMALL = .FALSE.
   17 IF( .NOT. RSMALL) GO TO 19
C*** ***SMALL ELEMENT -- FORM XIJ2D, YIJ2D, LIJ
C
C*** ***K5 = 105 FOR SMALL ELEMENT AXISYMMETRIC
      ASSIGN 105 TO K5
C***   ***K6 = 320 FOR SMALL ELEMENT   CROSS FLOW
      ASSIGN 320 TO K6
C*** ***NSW = 1  FOR I NE J
C*** ***NSW = 2  FOR I EQ J 1ST TIME THROUGH
C*** ***NSW = 3  FOR I EQ J 2ND TIME THROUGH
      GO TO (14, 21, 22), NSW
C
C*** ***I = J  1ST TIME THROUGH
   21 XLEFT = XJ + DX
      YLEFT = YJ + DY
      J1P1  = J + 1
      XRIGHT = X1(J1P1)
      YRIGHT = Y1(J1P1)
C*** ***GET NSW READY FOR I = J  2ND TIME THROUGH
      NSW = 3
      GO TO 23
C*** ***I = J   2ND TIME THROUGH
   22 XLEFT = X1(J1)
      YLEFT = Y1(J1)
      XRIGHT = XLEFT + 32.0 * DX
      YRIGHT = YLEFT + 32.0 * DY
      NSW = 1
C*** ***CALCULATE QUANTITIES WHICH HAVE NOT YET BEEN CALCULATED FOR I=4
   23 XMXJ = X2(I) - XLEFT
      YMYJ = Y2(I) - YLEFT
      XMXJP1 = X2(I) - XRIGHT
      YMYJP1 = Y2(I) - YRIGHT
      RLEFT = R
      RRIGHT = R
      S = SQRT( (XLEFT - XRIGHT)**2  + (YLEFT - YRIGHT)**2 )
      GO TO 11
C*** ***I NE J  SMALL ELEMENT
   14 RLEFT = R1SQR
      RRIGHT = R2SQR
C*** ***NOW FORM XIJ2D, YIJ2D, LIJ
   11 H = (-XMXJ * SINA(J) ) + ( YMYJ * COSA(J) )
      EL1 = (XMXJ * COSA(J) )+ ( YMYJ * SINA(J) )
      EL2 = (XMXJP1 * COSA(J) ) + (YMYJP1 * SINA(J) )
      DPHIDX =-ALOG (RLEFT / RRIGHT)
      IF(ABS(H/EL1) .LT. 10.0E-10)GO TO 7
      DPHIDY = -2.0 * ( ATAN(EL1/H) - ATAN(EL2/H) )
      GO TO 8
    7 DPHIDY = 0.0
      IF( (EL1 * EL2) .LT. 0.0) DPHIDY = -6.283186
    8 XIJ2D = (COSA(J)*DPHIDX) - (SINA(J)*DPHIDY)
      YIJ2D = (SINA(J)*DPHIDX) + (COSA(J)*DPHIDY)
      LIJ2D = ( (-(EL1 + EL2) / 4.0 ) * DPHIDX )
     1      + ( (S/4.0) * ALOG( (RLEFT *RRIGHT) / ( 4096.0 * T1**2) ) )
     2         -S  - ( (H/2.0) * DPHIDY )
C                * NO. OF INTERVAL LOOP
   19 DO 1000 IS=1,NI
      XJ=XJ+DX
      YJ=YJ+DY
      T2=YJ*YJ
      T3=X2(I)-XJ
      T4=T3*T3
      T5=(Y2(I)+YJ)**2
      T6=T4+T5
      T7=SQRT(T6)
      T8=T2+T4
      T9A = Y2(I) - YJ
      T9 = T9A**2                                                       XYZ2 141
      T10=T9+T4                                                         XYZ2 142
      T10A = SQRT(T10)                                                  XYZ2 143
C                                                                       XYZ2 144
C   ***   IF DENOM  (T8)   IS ZERO THEN MAKE T21 FAIL ALL TESTS         XYZ2 145
C                                                                       XYZ2 146
      IF( ABS(T8) .LT. 10.0E-30)GO TO 29                                XYZ2 147
      T21 = SQRT( T1 / T8 )                                             XYZ2 148
      GO TO 27                                                          XYZ2 149
   29 T21 = 0.10                                                        XYZ2 150
C                 * COMPUTE ELLIPIC INTEGRAL                            XYZ2 151
   27 IF(RSMALL  .AND.  FLG21 .EQ. 0)GO TO 18                           XYZ2 152
      XK=4.*YJ*Y2(I)/T6                                                 XYZ2 153
      CALL ELIP                                                         XYZ2 154
      IF ( IEC ) 18,575,18                                              XYZ2 155
   18 IF (IAC) 200,20,20                                                XYZ2 156
C                 * AXIS FLOW                                           XYZ2 157
   20 IF (RSMALL)GO TO 25                                               XYZ2 158
      T11 = YJ/T7                                                       XYZ2 159
      IF ( T21.LT.0.01) GO TO 24                                        XYZ2 160
      T12 = YJ/Y2(I)                                                    XYZ2 161
      FV2 = (EKK+EEK*(T1-T8)/T10)/T7                                    XYZ2 162
      FV3 = Y2(I)/T10 * T3/T7 * EEK                                     XYZ2 163
      F1 = FV3*T12                                                      XYZ2 164
      F2 = FV2*T12                                                      XYZ2 165
      FV4 = FV2*T3/Y2(I)                                                XYZ2 166
      F3=T11*EKK                                                        XYZ2 167
      GO TO 26                                                          XYZ2 168
   24 FV2 = 0.                                                          XYZ2 169
      FV3 = 0.                                                          XYZ2 170
      FV4 = 0.                                                          XYZ2 171
C***   ***SMALL Y FORMULAS AXISYMMETRIC FLOW                            XYZ2 172
      T23 = T1 / T8**2                                                  XYZ2 173
      T24 = 2.0 * T4 - T2                                               XYZ2 174
      F1 = ( (   1.570796  * YJ * T3 ) / ( T8**1.5      ) ) *           XYZ2 175
```

```
     1  ( 1.0 + ( .75 * ( 3.0 * T2 - 2.0 * T4 ) * T23 ) )
      F2 = (   1.570796   * YJ * Y2(I) ) * ( T24 / (T8**2.5       ) )
      F3 = 1.570796 * YJ * ( 1.0 + (.25 * T23 * (-T24) ) ) / SQRT(T8 )
      GO TO 26
   25 T32 = T3 / T10A
      T33 = T9A / T10A
      T34 = T33**2
      T35A = T10A / (8.0 * Y2(I) )
      T35 = ALOG(T35A)
      T36 = T9A/Y2(I)
      T40  = T10A / Y2(I)
      T37 = (T40**2)*0.125
      T38 = 0.250*T36*T35
      T39 = 0.125*T36
      T34A = 2.0*T34
      T34B = T34A + 3.0
      F1 = ( -2.0 * T32 * ( (-T35A * T35) - (0.5 * T33)
     1     -( (T40/16.0) * T34B ) ) ) / Y2(I)
      F2 =-( (0.25 * T36 * T35) -T34 -1.0 - (T39 * T34B) ) / Y2(I)
      F3 = ( T35 * ( T36 + (0.25 * T36**2) + T37 ) ) -T36 +T37
   26 GO TO K2, (28,110)
C              * SIMPSON RULE INTEGRATION
   28 IF (IS-1) 30,30,40
C              * FIRST PASS
   30 AXS=F1
      AYS=F2
      AZS=F3
      IA=0
      GO TO 110
   40 IF(IS.EQ.NI)GO TO 75
   50 IF (IA) 70,60,70
C              * EVEN PASS
   60 AXS=AXS+4.*F1
      AYS=AYS+4.*F2
      AZS=AZS+4.*F3
      IA=1
      GO TO 110
C                * ODD PASS
   70 AXS=AXS+F1+F1
      AYS=AYS+F2+F2
      AZS=AZS+F3+F3
      IA=0
      GO TO 110
   75 GO TO KS, (80,105)
C                * LAST PASS
   80 IF (J-I) 100,90,100
   90 IF (BON.NE.0.0) GO TO 100
      AX(J)=AX(J)-S4*(AXS+F1)
      AY(J)=AY(J)-S2*(AYS+F2)
      AZ(J)=AZ(J)+S4*(AZS+F3)
      GO TO 110
  100 AX(J)=-S4*(AXS+F1)
      AY(J)=-S2*(AYS+F2)
      AZ(J)=S4*(AZS+F3)
      GO TO 110
C*** ***LAST PASS  *  SMALL ELEMENT
  105 IF( (J .NE.I) .OR. (BON .NE. 0.0) )GO TO 107
C*** ***I = J   ON BODY
      AX(J) = AX(J) + XIJ2D + (AXS + F1) * S1
      AY(J) = AY(J) + YIJ2D + (LIJ2D / Y2(I) ) + (AYS + F2) * S1
      AZ(J) = AZ(J) -(2.0 * LIJ2D) + (AZS + F3) * S1
      GO TO 110
C*** ***I NE J   ON OR OFF BODY
  107 AX(J) = XIJ2D  + (AXS + F1) * S1
      AY(J) = YIJ2D + (LIJ2D / Y2(I) ) + (AYS + F2) * S1
      AZ(J) = -2.0 * LIJ2D + (AZS + F3)  * S1
  110 IF (IAC) 200,200,400
C                * CROSS FLOW
  200 IF (RSMALL)GO TO 223
      IF (T21 .LT. 0.04)GO TO 220
      T12 = T1 + T8
      F1=T3/Y2(I)*(EKK-EEK*T12/T10)/T7
      F2=(EEK*(T8*T8+T1*(T4-T2))/T10-EKK*T8)/T1/T7
      F3=T7*(EKK*T12/T6-EEK)/T1
      GO TO 230
C***   ***SMALL Y FORMULAS    *    CROSS FLOW
  220 T23 = T1 / T8**2
      T29 = ( 1.570796 * T2 ) / ( T8**1.5       )
      T26 = 4.0 * T4 - T2
      T31 = T26 * T23
      F1 = ( (-4.712389) * T2 * T3 * Y2(I) ) / ( T8**2.5         )
      F2 = T29 * ( 1.0 - (1.125 * T31 ) )
      F3 = T29 * ( 1.0 - (.375 * T31) )
      GO TO 230
C*** ***IAC LT 0 MEANS NO AXISYMMETRIC FLOW
  223 IF(IAC)225,227,227
C*** ***CALCULATE SMALL ELEMENT QUANTITIES THAT DID NOT GET CALCULATED
C*** ***BECAUSE THERE WAS NO AXISYMMETRIC FLOW
  225 T32 = T3 / T10A
      T33 = T9A / T10A
      T34 = T33**2
      T35A = T10A / (8.0 * Y2(I) )
      T35 = ALOG(T35A)
      T36 = T9A/Y2(I)
      T40  = T10A / Y2(I)
      T37 = (T40**2)*0.125
      T38 = 0.250*T36*T35
C*** ***CALCULATE SMALL ELEMENT F1,F2,F3  CROSS FLOW
  227 T38A = -5.0 * T38
      T40A = T40**2
      T36A = T36**2
      F1 = (T32 / Y2(I)) * ( (-0.75 * T40 * T35) + T33
     1   +(0.125 * T40 * (2.0 * T34 -5.0) ) )
      F2 = ( T38A + T34 + 3.0 + (0.25 * T36 * (T34 - 6.50) ) ) / Y2(I)
      F3 = ( ( T36 - (0.375 * T40A) - (0.25 * T36A) ) * T35 -4.0 +T36
     1      -(0.50 * T36A) - T37 ) / Y2(I)
C                  * SIMPSON RULE INTEGRATION
  230 IF (IS-1) 240,240,250
C                  * FIRST PASS
  240 CXS=F1
      CYS=F2
      CZS=F3
      IC=0
      GO TO 400
  250 IF (IS-NI) 260,290,260
  260 IF (IC) 280,270,280
C                  * EVEN PASS
  270 CXS=CXS+4.*F1
      CYS=CYS+4.*F2
      CZS=CZS+4.*F3
      IC=1
      GO TO 400
C                  * ODD PASS
  280 CXS=CXS+F1+F1
      CYS=CYS+F2+F2
      CZS=CZS+F3+F3
      IC=0
      GO TO 400
  290 GO TO K6,(295,320)
C                  * LAST PASS
  295 IF(J .NE. I)GO TO 310
  300 IF (RON.NE.0.0) GO TO 310
      CX(J)=CX(J)+S2*(CXS+F1)
      CY(J)=CY(J)+S2*(CYS+F2)
      CZ(J)=CZ(J)+S2*(CZS+F3)
      GO TO 400
  310 CX(J)=S2*(CXS+F1)
      CY(J)=S2*(CYS+F2)
      CZ(J)=S2*(CZS+F3)
      GO TO 400
  320 IF ( (I.NE.J) .OR. (BON .NE. 0.0) ) GO TO 340
C*** ***LAST PASS   SMALL ELEMENT   I=J  ON BODY
      CX(J) = CX(J) + XIJ2D + (CXS + F1) * S1
      CY(J) = CY(J) + YIJ2D + (LIJ2D / Y2(I)) + (CYS + F2) * S1
      CZ(J) = CZ(J) -(2.0 * LIJ2D / Y2(I) )    + (CZS + F3) * S1
      GO TO 400
C*** ***I NE J OR ANY OFF BODY
  340 CX(J) = XIJ2D + (CXS + F1) * S1
      CY(J) = YIJ2D + (LIJ2D/Y2(I)) +(CYS + F2) * S1
      CZ(J) = -(2.0 * LIJ2D / Y2(I) ) + (CZS + F3) * S1
C*** ***K3 = 420 FOR SURFACE VORTICITY   PF TRUE
  400 GO TO K3,(410,420)
C*** ***K1 = 420 FOR STRIP VORTEX
  410 GO TO K1, (570,420)
C*** ***FLOW OF CONTROL REACHES HERE FOR (PF=TRUE) OR ( (FLG15 GT 0 AND
C*** ***NLF LE 0 (LIFTING BODY)) AND (I NE J ON BODY OR ANY OFF BODY) )
  420 IF (RSMALL)GO TO 542
      FV1 = (T2-T1) / T7  * EEK / T10
      IF (IS.GT.1) GO TO 440
C                  * FIRST PASS
      AX1 = FV1
      AX2 = FV2
      AY1 = FV3
      AY2 = FV4
      IV=0
      GO TO 570
  440 IF (IS.EQ.NI) GO TO 500
      IF (IV) 460,450,460
C                  * EVEN PASS
  450 AX1 = AX1+4.*FV1
      AX2 = AX2+4.*FV2
      AY1 = AY1+4.*FV3
      AY2 = AY2+4.*FV4
      IV=1
      GO TO 570
```

```
C                 * ODD PASS
  460 AX1 = AX1+FV1+FV1
      AX2 = AX2+FV2+FV2
      AY1 = AY1+FV3+FV3
      AY2 = AY2+FV4+FV4
      IV = 0
      GO TO 570
C                 * LAST PASS
  500 IF (J-I) 540,520,540
  520 IF (BON.NE.0.) GO TO 540
      AXV(J) = AXV(J) - S4*(AX1+FV1) - S2*(AX2+FV2)
      AYV(J) = AYV(J) - S4*(AY1+FV3) + S2*(AY2+FV4)
      GO TO 550
  540 AXV(J) = -S4*(AX1+FV1) -S2*(AX2+FV2)
      AYV(J) = -S4*(AY1+FV3) +S2*(AY2+FV4)
      GO TO 550
  542 T34C = T34B - 8.0
      FV1 = ( T38 + (T32**2) - (T39 * T34C) ) / Y2(I)
      FV2 = ( T32 / Y2(I) ) * ( (-0.75 * T40 * T35) +T33
     1      +(0.125 * T40 * T34C) )
      IF (IS .GT.1)GO TO 544
C*** ***FIRST PASS  SMALL ELEMENT
      AX1 = FV1
      AY1 = FV2
      IV = 0
      GO TO 570
  544 IF(IS .EQ. NI)GO TO 548
      IF(IV .NE. 0 )GO TO 546
C*** ***EVEN PASS   SMALL ELEMENT
      AX1 = AX1 + 4.0* FV1
      AY1 = AY1 + 4.0* FV2
      IV = 1
      GO TO 570
C*** ***ODD PASS    SMALL ELEMENT
  546 AX1 = AX1 + FV1 + FV1
```

```
      AY1 = AY1 + FV2 + FV2                                             XYZ2  386
      IV = 0                                                            XYZ2  387
      GO TO 570                                                         XYZ2  388
C*** ***LAST PASS   SMALL ELEMENT                                       XYZ2  389
  548 IF ( (I .NE. J) .OR. (BON .NE. 0.0) )GO TO 549                    XYZ2  390
C*** ***I = J   ON  BODY                                                XYZ2  391
      AXV(J) = AXV(J) - YIJ2D + (LIJ2D / Y2(I) ) + (AX1 + FV1)*S1       XYZ2  392
      AYV(J) = AYV(J) + XIJ2D + (AY1 + FV2)*S1                          XYZ2  393
      GO TO 550                                                         XYZ2  394
C*** ***I NE J  OR  ANY OFF BODY                                        XYZ2  395
  549 AXV(J) = -YIJ2D  + (LIJ2D / Y2(I) ) + (AX1 + FV1)*S1              XYZ2  396
      AYV(J) = XIJ2D +(AY1 + FV2)*S1                                    XYZ2  397
C*** ***K4 = 560 FOR SURFACE VORTICITY   PF TRUE                        XYZ2  398
  550 GO TO K4,(560,570)                                                XYZ2  399
C*** ***FLOW OF CONTROL REACHES HERE IF PF IS TRUE                      XYZ2  400
  560 AX(J) = AXV(J)                                                    XYZ2  401
      AY(J) = AYV(J)                                                    XYZ2  402
  570 IF (IEC.EQ.-1) GO TO 1000                                         XYZ2  403
  575 IF (T21.LT.0.08)GO TO 595                                         XYZ2  404
  580 T20 = SQRT( T2 / (T1 + T4) )                                      XYZ2  405
      IF (T20.LT.0.01) GO TO 590                                        XYZ2  406
      T13 = YJ * Y2(I)**3                                               XYZ2  407
      T14 = T1 + T8                                                     XYZ2  408
      T15 = T2 * T1                                                     XYZ2  409
      T16 = T14 * T14                                                   XYZ2  410
      T17 = T1 * YJ                                                     XYZ2  411
      T18 = T1 * T1                                                     XYZ2  412
      T19 = T8 * T8                                                     XYZ2  413
      F3 = ( T7/T13 ) * ( (-T14) * EFK + ( ( (T16 - T15) * EKK) / T6 ) )  XYZ2  414
      F1 =(T3 / (T17 *T7) ) * ( (EFK / T10 ) * (T16 - 3.0 * T15) -     XYZ2  415
     1  (T14 * EKK) )                                                   XYZ2  416
      TEMP1 =((-8.0*T8**3) - (12.0*T1*T19) + (26.0*T15*T8)             XYZ2  417
     1    + (2.0*T18*(2.0*T1 -5.0*T2) ))* EFK/T10                       XYZ2  418
      TEMP2 = EKK * ( (8.0*T19) + (4.0*T1*T8) - (2.0*T15) - (4.0*T18) )  XYZ2  419
      F2 = (TEMP1 + TEMP2) / ( T13 * T7)                                XYZ2  420
      GO TO 630
C***   ***SMALL YJ FORMULAS   *   FXTRA CROSS FLOW
  590 T25 = YJ**3
      T30 = T4 + T1
      T27 = T30**3.5
      T28 = T25 * Y2(I)
      F1 = (    2.945243   * T25 * T3 * T1) / T27
      F2 =    7.068584   * T28 * ( 3.0 * T1   -  2.0 * T4 ) / T27
      F3 = 1.767146 * T28 /(T30**2.5     )
      GO TO 630
C***   ***SMALL Y FORMULAS   *   EXTRA CROSS FLOW
  595 T25 =  YJ**3
      F1 = (    2.945243   * T25 * T3 * T1 ) / ( T8**3.5        )
      F2 = ( (-14.13717) * T25 * Y2(I) ) / (T8**2.5)
      F3 = -F2 / 8.0
C***   ***SIMPSON*S RULE
  630 IF (IS - 1) 640,640,650
C***   ***FIRST PASS
  640 ECXS = F1
      ECYS = F2
      ECZS = F3
      IE = 0
      GO TO 1000
  650 IF (IS - NI) 660,690,660
  660 IF ( IE ) 680,670,680
C***   ***FVEN PASS
  670 FCXS = ECXS + 4.0 * F1
      ECYS = ECYS + 4.0 * F2
      FCZS = ECZS + 4.0 * F3
      IE = 1
      GO TO 1000
C***   ***ODD PASS
  680 ECXS = ECXS + F1 + F1
      FCYS = ECYS + F2 + F2
      FC7S = ECZS + F3 + F3
      IE = 0
      GO TO 1000
C***   ***LAST PASS
  690 IF(J - I) 710,700,710
C***   ***I=J      *       ELEMENTS ON MAIN DIAGONAL
  700 IF (BON.NE.0.0) GO TO 710
      ECX(J) = ECX(J) -S4 * ( ECXS + F1 )
      ECY(J) = ECY(J) -S5 * ( ECYS + F2 )
      ECZ(J) = ECZ(J) +S3 * ( ECZS + F3 )
      GO TO 1000
C***   ***OFF MAIN DIAGONAL OR OFF BODY POINTS
  710 ECX(J) = -S4 * ( ECXS + F1 )
      ECY(J) = -S5 * ( ECYS + F2 )
      ECZ(J) =  S3 * ( ECZS + F3 )
 1000 CONTINUE
      RETURN
      END
      SUBROUTINE ELIP                                                     ELIP  001
C                                                                         ELIP  002
C              * HASTINGS APPROXIMATION FOR FLLIPTIC INTERGALS            ELIP  003
C                                                                         ELIP  004
      COMMON      HEDR(10)  ,CASE      ,NB        ,NNU                    ELIP  005
     1            ,FLG03    ,FLG04     ,FLG05     ,FLG06     ,FLG07       ELIP  006
     2            ,FLG08    ,FLG09     ,FLG10     ,FLG11     ,FLG12       ELIP  007
     3            ,FLG13    ,FLG14     ,FLG15     ,FLG16     ,FLG17       ELIP  008
     4            ,FLG18    ,FLG19     ,FLG20     ,FLG21     ,FLG22       ELIP  009
     5            ,FLG23    ,FLG24     ,FLG25     ,FLG26     ,FLG27       ELIP  010
      COMMON   NT,        ND(11),    MN,        NUNA(5),  TYPEA(5),       ELIP  011
     1         NER1,      NER2,      NMA,       NSIGA,    NSIGC,          ELIP  012
     2         NUNC(5),   TYPEC(5),  NLF(11),   IEC,      NSIGEC,         ELIP  013
     3         TYPEEC(5),NUNEC(5)                                         ELIP  014
C     DOUBLE PRECISION  HEDR, CASE                                        ELIP  015I
      INTEGER      FLG03    ,FLG04     ,FLG05     ,FLG06     ,FLG07       ELIP  016
     1            ,FLG08    ,FLG09     ,FLG10     ,FLG11     ,FLG12       ELIP  017
     2            ,FLG13    ,FLG14     ,FLG15     ,FLG16     ,FLG17       ELIP  018
     3            ,FLG18    ,FLG19     ,FLG20     ,FLG21     ,FLG22       ELIP  019
     4            ,FLG23    ,FLG24     ,FLG25     ,FLG26     ,FLG27       ELIP  020
      REAL         MN                                                     ELIP  021
      LOGICAL PF                                                          ELIP  022
C                                                                         ELIP  023
      COMMON /CL/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),    ELIP  024
     1              SINA(100),COSA(100),XP(100),  YP(100)                 ELIP  025
     2              XWAKE(11),YWAKE(11)                                   ELIP  026
      COMMON /TL/   A(100),   B(100),   AX(100),  AY(100),  AZ(100),      ELIP  027
     1              CX(100),  CY(100),  CZ(100),  AXV(100),AYV(100),      ELIP  028
     2              VN(100,5),VT(100,5),BUN,           IAC,               ELIP  029
     3              I,        J,        J1,       SJ,       DS,           ELIP  030
     4              DX,       DY,       NI,       XJ,       YJ,           ELIP  031
     5              XK,       EEK,      EKK,      K,        PF            ELIP  032
C                                                                         ELIP  033
C              * START                                                    ELIP  034
      ETA = 1. - XK                                                       ELIP  035
      IF (FTA) 20,20,40                                                 ELIP  036
   20 WRITF (6,30) ETA                                                  ELIP  037
   30 FORMAT ( 1H0 36H *** ERROR IN SUBROUTINE ELIP * ETA= F15.8 )      ELIP  038
       WRITF(6,800) I,XJ,DX,YJ,DY,X2(I),Y2(I),XK                        ELIP  039
  800  FORMAT(1H ,I5,7F15.6)                                            ELIP  040
       FTA = 0.000005                                                   ELIP  041
   40 ELN=ALOG(ETA)                                                     ELIP  042
      FKK = 1.386294F0      + FTA * (.9666344E-1    + ETA *             ELIP  043
     1      (.3590092E-1    + ETA * (.3742564E-1    + ETA *             ELIP  044
     2      .1451196F-1   ))) - FLN * (.5 + ETA * (.1249859E0     +     ELIP  045
     3      FTA * (.6880249E-1    + ETA * (.3328355E-1    + ETA *       ELIP  046
     4      .4417870E-2  ))))                                           ELIP  047
      FEK = 1. + FTA * (.4432514F0      + ETA * (.6260601E-1    + FTA * ELIP  048
     1      (.4757384E-1    + ETA * .1736506F-1   ))) - ELN * (ETA *    ELIP  049
     2      (.2499837E0    + ETA * (.9200180F-1    + ETA *              ELIP  050
     3      (.4069698E-1    + ETA * .5264496F-2  ))))                   ELIP  051
      RETURN                                                            ELIP  052
      FND                                                               ELIP  053
      SUBROUTINE NOTS                                                    NOTS  001
      COMMON      HEDR(10)  ,CASE      ,NB        ,NNU                   NOTS  002
     1            ,FLG03    ,FLG04     ,FLG05     ,FLG06     ,FLG07      NOTS  003
     2            ,FLG08    ,FLG09     ,FLG10     ,FLG11     ,FLG12      NOTS  004
     3            ,FLG13    ,FLG14     ,FLG15     ,FLG16     ,FLG17      NOTS  005
     4            ,FLG18    ,FLG19     ,FLG20     ,FLG21     ,FLG22      NOTS  006
     5            ,FLG23    ,FLG24     ,FLG25     ,FLG26     ,FLG27      NOTS  007
C                                                                        NOTS  008
      COMMON    NT,        ND(11),    MN,        NUNA(5),   TYPEA(5),    NOTS  009
     1          NER1,      NER2,      NMA,       NSIGA,     NSIGC,       NOTS  010
     2          NUNC(5),   TYPEC(5),  NLF(11),   IEC,       NSIGEC,      NOTS  011
     3          TYPEEC(5),NUNEC(5)                                       NOTS  012
C     DOUBLE PRECISION HEDR,CASE                                         NOTS  013I
C                                                                        NOTS  014
      COMMON   /RNGWNG/ VA(100,2),  VR(100,2),VAN(100), VAT(100)         NOTS  015
C                                                                        NOTS  016
      INTEGER      FLG03    ,FLG04     ,FLG05     ,FLG06     ,FLG07      NOTS  017
     1            ,FLG08    ,FLG09     ,FLG10     ,FLG11     ,FLG12      NOTS  018
     2            ,FLG13    ,FLG14     ,FLG15     ,FLG16     ,FLG17      NOTS  019
     3            ,FLG18    ,FLG19     ,FLG20     ,FLG21     ,FLG22      NOTS  020
     4            ,FLG23    ,FLG24     ,FLG25     ,FLG26     ,FLG27      NOTS  021
C                                                                        NOTS  022
      COMMON /CL/    X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),  NOTS  023
     1               SINA(100),COSA(100),XP(100),  YP(100)               NOTS  024
     2              ,XWAKE(11),YWAKE(11)                                 NOTS  025
C                                                                        NOTS  026
      COMMON /TL/    A(100),   B(100),   AX(100),  AY(100),  AZ(100),    NOTS  027
     1               CX(100),  CY(100),  CZ(100),  AXV(100),AYV(100),    NOTS  028
     2               VN(100,5),VT(100,5),BUN,                IAC,        NOTS  029
     3               I,        J,        J1,       SJ,       DS,         NOTS  030
     4               DX,       DY,       NI,       XJ,       YJ,         NOTS  031
     5               XK,       EEK,      EKK,      K,        PF          NOTS  032
C                                                                        NOTS  033
      REAL  MN,KAY                                                       NOTS  034
      LOGICAL PF                                                         NOTS  035
```

```
C                                                                       NOTS  036
C*** *  FOLLOWING ARE 3 ARITHMETIC FUNCTIONS                            NOTS  037
C                                                                       NOTS  038
      OMEG(Z,SMALLR,BIGR) = 1.0 + ( ( Z**2 + (SMALLR-BIGR)**2) /        NOTS  039
     1                      (2.0*SMALLR * BIGR) )                       NOTS  040
      BETAF(Z,SMALLR,BIGR)= ARSIN( Z / ( SQRT(Z**2 + (SMALLR-BIGR)**2)))NOTS  041
C                                                                       NOTS  042
C                                                                       NOTS  043
      AKAYF(Z,SMALLR,BIGR)= SQRT( (4.0 * SMALLR * BIGR) /               NOTS  044
     1                      ( Z**2 + (SMALLR + BIGR)**2 ) )             NOTS  045
C                                                                       NOTS  046
      DO 100 IBOD =1,FLG14                                              NOTS  047
      Z = X2(I) - XWAKE(IBOD)                                           NOTS  048
      OMEGA = OMEG( Z, Y2(I), YWAKE(IBOD) )                             NOTS  049
      BETA  =BETAF( Z, Y2(I), YWAKE(IBOD) )                             NOTS  050
      KAY   =AKAYF( Z, Y2(I), YWAKE(IBOD) )                             NOTS  051
      CALL QC(OMEGA,QM,Q)                                               NOTS  052
C                                                                       NOTS  053
C   ***   SMALLR IS Y2(I)                                               NOTS  054
C   ***   BIGR IS YWAKE(IBOD)                                           NOTS  055
      IF( Y2(I) .LE. YWAKE(IBOD) ) GO TO 30                             NOTS  056
C                                                                       NOTS  057
C   ***   SMALLR GT BIGR                                                NOTS  058
C                                                                       NOTS  059
      BIGK = ( Z / ( SQRT( Y2(I) * YWAKE(IBOD) )*2.0 ) ) * QM -         NOTS  060
     1        ( 1.570796 * HLAMB(BETA,KAY) )                            NOTS  061
      GO TO 40                                                          NOTS  062
C                                                                       NOTS  063
C*** *  SMALLR LE BIGR                                                  NOTS  064
C                                                                       NOTS  065
   30 BIGK = 3.141593 + ( Z / ( SQRT( Y2(I) * YWAKE(IBOD) )*2.0) ) *QM +NOTS  066
     1        ( 1.570796 * HLAMB(BETA,KAY) )                            NOTS  067
C                                                                       NOTS  068
C*** *  NOTE THAT VA AND VR WILL NOT YET BE MULTIPLIED BY  DGAMMA/DZ    NOTS  069
C*** *  WHICH IS REALLY THE INPUT PRESCRIBED VORTICITY                  NOTS  070
```

```
C                                                                          NOTS  071
   40 VA(I,IHOD) = BIGK / 6.2831A5                                         NOTS  072
  100 VR(I,IBOD) = -(Q * (SQRT(YWAKE(IBOD) / Y2(I) ) ) )  /  6.283185      NOTS  073
      RETURN                                                               NOTS  074
      END                                                                  NOTS  075
      FUNCTION HLAMB (BETA,K)                                           HLAB  001
C  THIS SURRROUTINE CALCULATES THE HEUMAN≠S LAMBDA FUNCTION OF BETA AND K HLAB  002
C     DOUBLE PRECISION A,F,E                                            HLAB  003I
      REAL K                                                            HLAB  004
      DATA TWOP/0,6366197724/                                           HLAB  005
      CALL INEL (FI,EI,PI,BETA,BETA,1,0-K**2       ,0,1,1)              HLAB  006
      A = 1,0 - K  **2                                                  HLAB  007
      CALL ELLC (A          ,F,E,1)                                     HLAB  008
      CALL ELLC (A          ,F,E,2)                                     HLAB  009
      HLAMB = TWOP*(F*EI +(E-F)*FI)                                     HLAB  010
      RETURN                                                            HLAB  011
      END                                                               HLAB  012
      SUBROUTINE QC(OMEG,QM,Q)
C  THIS SUBROUTINE CALCULATES THE LEGENDRE FUNCTIONS OF THE SECOND KIND
C  AND HALF ORDER, THE ARGUMENTS ARE=
C        OMEG    ARGUMENT FOR WHICH LEGENDRE FUNCTIONS WILL BE FOUND
C        QM      VALUE OF LEGENDRE FUNCTION OF MINUS ONE HALF ORDER
C        Q       VALUE OF LEGENDRE FUNCTION OF PLUS ONE HALF ORDER
C     DOUBLE PRECISION OMEGD,ARG,A,F,E,QMD,QD
      OMEGD=OMEG
      ARG=2.0/(OMEGD+1.0)
      A=1.0-ARG
      CALL ELLC (A,F,E,1)
      CALL ELLC (A,F,E,2)
      QMD=F*ARG**0.5
      QD=-E*(2.0*(OMEGD+1.0))**0.5+OMEGD*QMD
      QM=QMD
      Q=QD
      RETURN
      END
      SUBROUTINE ELLC (A,K,E,I)                                         FLLC  001
C   THIS SUBROUTINE CALCULATES THE ASSOCIATED COMPLETE ELLIPTIC INTEGRALS ELLC  002
C   OF THE FIRST OR SECOND KIND                                         ELLC  003
C   THE ARGUMENTS ARE≠                                                  ELLC  004
C        A   ARGUMENT (K SQUARED) FOR WHICH E≠ OR K≠ WILL BE FOUND      ELLC  005
C        K   VALUE OF ASSOCIATED COMPLETE ELLIPTIC INTEGRAL OF FIRST KIND ELLC  006
C        E   VALUE OF ASSOCIATED COMPLETE ELLIPTIC INTEGRAL OF SECOND KIN ELLC  007
C        I   IF EQ 1, COMPUTE K ; IF EQ 2, COMPUTE E                    FLLC  008
C     DOUBLE PRECISION K,E,CON(32),A,LN4,CF(29),CL(3),DLOG              ELLC  009I
      DOUBLE PRECISION CON(32),CF(29),CL(3)                             ELLC  010C
      EQUIVALENCE (CON,CF),(CON(30),CL)                                 ELLC  011
      DATA CF  /9.6573590797589018D-2,3.0885573486752694D-2,1.4978988178 FLLC  012
     17046290-2,9.6587579861753113D-3,1.1208918554644092D-2,1.3855601247 ELLC  013
     215656D-2,6.6905509906897936D-3,6.4998443329390180D-4,1.249999999411 ELLC  014
     37923D-1,7.0312426464627361D-2,4.8818058565403952D-2,3.706839893415 ELLC  015
     45422D-2,2.7189861116788250D-2,1.4105380776158048D-2,3.1831309927862 ELLC  016
     5886D-3,1.5049181783601883D-4,4.4314718112155806D-1,5.6805657874695 ELLC  017
     6358D-2,2.1876220647186198D-2,1.2510592410844644D-2,1.3034146073731 ELLC  018
     7432D-2,1.5377102528552019D-2,7.3356164974290365D-3,7.0980964089987 FLLC  019
     8229D-4,2.4999999936176222D-1,9.3749920249680113D-2,5.8582839536559 FLLC  020
     9024D-2,4.2382807456947 9D-2,3.0302747728412848D-2 /               ELLC  021
      DATA CL  / 1.5525129948040721D-2,3.4838679435896492D-3,1.642721079 ELLC  022
     17048025D-4 /                                                      ELLC  023
      LN4 = 1.3862943611198900                                          ELLC  024
      IF (A.EQ.0.0) GO TO 4                                             ELLC  025
      GO TO (1,2),I                                                     ELLC  026
    1 K = LN4 + (((((((CON(8)*A+CON(7))*A+CON(6))*A+CON(5))*A+CON(4))*A FLLC  027
     1+ CON(3))*A+CON(2))*A+CON(1))*A - DLOG(A)*(0.5+(((((((CON(16)*A+  ELLC  028
     2CON(15))*A+CON(14))*A+CON(13))*A+CON(12))*A+CON(11))*A+CON(10))*A ELLC  029
     3+ CON(9))*A)                                                      ELLC  030
      GO TO 3                                                           ELLC  031
    2 E = 1.0D0+(((((((CON(24)*A+CON(23))*A+CON(22))*A+CON(21))*A+CON(20 ELLC  032
     1))*A+CON(19))*A+CON(18))*A+CON(17))*A - DLOG(A)*((((((((CON(32)*A ELLC  033
     2+ CON(31))*A+CON(30))*A+CON(29))*A+CON(28))*A+CON(27))*A+CON(26))* ELLC  034
     3A+CON(25))*A)                                                     ELLC  035
```

```
      3 RETURN                                                          ELLC  036
C     4 K = 0.99999999D30                                               ELLC  037I
      4 K = 0.99999999F30                                               ELLC  038C
C       F = 1.0D0                                                       ELLC  039I
        F = 1.0E0                                                       ELLC  040C
        RETURN                                                          ELLC  041
        END                                                             ELLC  042
      SUBROUTINE ELINT3(XKSQ,XN,PHI,PIE)
C  THIS SUBROUTINE CALCULATES THE INCOMPLETE ELLIPTIC INTEGRAL OF THE
C  THIRD KIND. THE ARGUMENTS ARE≠
C        XKSQ    VALUE OF K SQUARED
C        XN      VALUE OF MINUS ALPHA SQUARED
C        PHI     VALUE OF PHI
C        PIE     VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THIRD KIND
  101 FORMAT (7E16.8)
      DATA HP /1.570796/
      DATA ROUND /.0000050/
      SK=XKSQ
      FN=XN
      P=PHI
      IF (FN.EQ.-1.0.AND.SK.EQ.1.0) GO TO 50
      IF(SK.GT.1.) GO TO 48
      IF(FN.LT.(-1.)) GO TO 48
      IF(P)1,48,2
C     NORMALIZE PHI
 1    A=-1.
      P=-P
      GOTO3
 2    A=1.
 3    B=1.
      BB=1.
      IF (ABS(P-1.570796   ).LE.10.0**(-7)) GO TO 10
      IF(P-HP)11,10,4
 4    J=P/(2.*HP)
      XX=2*J
      P1=P-XX*HP
      P=HP
      B=-1.
      GOTO10
 5    D=SUM
      B=0.
      IF(P1-HP)6,7,8
```

```
6       P=P1                                                            ELNT  036
        XXX=1.                                                          ELNT  037
        GOTO11                                                          ELNT  038
7       PIE=(XX+1.)*A*D                                                 ELNT  039
        GOTO47                                                          ELNT  040
8       XXX=-1.                                                         ELNT  041
        XX=XX+2.                                                        ELNT  042
        P=2.*HP-P1                                                      ELNT  043
        GOTO11                                                          ELNT  044
9       PIE=A*(XX*D+XXX*SUM)                                            ELNT  045
        GOTO47                                                          ELNT  046
10      IF(SK.EQ.1.) GOTO48                                             ELNT  047
        IF(FN.EQ.(-1.)) GO TO 48                                        ELNT  048
11      IF(P.GT.10.E-4)GOTO13                                           ELNT  049
        IF(FN.GT.0.)GOTO12                                              ELNT  050
        SUM=P                                                           ELNT  051
        GOTO45                                                          ELNT  052
12      RRT=SQRT(FN)                                                    ELNT  053
        SUM=ATAN(P*RRT)/RRT                                             ELNT  054
        GOTO45                                                          ELNT  055
13      S=SIN(P)                                                        ELNT  056
        S2=S**2                                                         ELNT  057
        C=COS(P)                                                        ELNT  058
        IF(SK.GT.0.64)GOTO20                                            ELNT  059
        IF(ABS(FN).GE.0.6)GOTO15                                        ELNT  060
C       POWER SERIES IN N AND K SQUARED                                 ELNT  061
        SA=1.                                                           ELNT  062
        SB=SK/2.                                                        ELNT  063
        CB=S*C                                                          ELNT  064
        CA=P                                                            ELNT  065
        FM=0.                                                           ELNT  066
        SUM=P                                                           ELNT  067
        X=SUM*1.E-8                                                     ELNT  068
14      SA=SB-SA*FN                                                     ELNT  069
        CA=(-CB/(2.*(FM+1.)))+(1.-.5/(FM+1.))*CA                        ELNT  070
```

```
      Y=SA*CA                                                           ELNT  071
      SUM=SUM+Y                                                         ELNT  072
      IF((SB*CA).GT.X) GO TO 141                                        ELNT  073
      IF(ABS(Y) .LT.X) GO TO 45                                         ELNT  074
 141  FM=FM+1.                                                          ELNT  075
      CB=CB*S2                                                          ELNT  076
      SB=(1.-.5/(FM+1.))*SK*SB                                          ELNT  077
      GOTO14                                                            ELNT  078
C     POWER SERIES IN K SQUARED                                         ELNT  079
 15   PK=SK                                                             ELNT  080
      RT=SQRT(1.+FN)                                                    ELNT  081
      IF(RT.NE.0.) GO TO 16                                             ELNT  082
      G=S/C                                                             ELNT  083
      GOTO18                                                            ELNT  084
 16   IF(C.GT.4.E-3)GOTO17                                              ELNT  085
      G=(HP-(C/(RT*S)))/RT                                              ELNT  086
      GOTO18                                                            ELNT  087
 17   G=ATAN(RT*S/C)/RT                                                 ELNT  088
 18   G1=G*1.E-8                                                        ELNT  089
      E=P                                                               ELNT  090
      F=S*C                                                             ELNT  091
      H=1.                                                              ELNT  092
      SUM=G                                                             ELNT  093
      FM=0.                                                             ELNT  094
 19   G=(E-G)/FN                                                        ELNT  095
      H=H*(1.-0.5/(FM+1.))                                              ELNT  096
      G2=H*G*PK                                                         ELNT  097
      SUM=SUM+G2                                                        ELNT  098
      IF(G2.LE.G1)GOTO45                                                ELNT  099
      FM=FM+1.                                                          ELNT  100
      E=-F/(2.*FM)+(1.-0.5/FM)*E                                        ELNT  101
      F=F*S2                                                            ELNT  102
      PK=PK*SK                                                          ELNT  103
      GOTO19                                                            ELNT  104
 20   SKP=1.-SK                                                         ELNT  105
```

```
      IF(S.LT.C) GO TO 32                                               ELNT 106
C     ADDITION FORMULA                                                  ELNT 107
 21   ZP=SQRT(1.-SK*S2)                                                 ELNT 108
      RT1=SQRT(ABS(FN*(FN+1.)*(FN+SK)))                                 ELNT 109
      SST=(1.-ZP)/(SK*(C+1.))                                           ELNT 110
      XP=(SST*S*RT1)/(1.+FN*S2-FN*SST*C*ZP)                             ELNT 111
      IF(FN)22,29,25                                                    ELNT 112
 22   IF(RT1.NE.0.) GO TO 24                                            ELNT 113
      R=S/(C+1.)                                                        ELNT 114
      IF(FN.NE.(-1.)) GO TO 23                                          ELNT 115
      CF=(2.*R-SK*SST*S-(S/C)*ZP)/SKP                                   ELNT 116
      GOTO30                                                            ELNT 117
 23   CF=(SST*(S-(2./R))+S*C/ZP)*(SK/SKP)                               ELNT 118
      GOTO30                                                            ELNT 119
 24   IF(FN*(FN+SK) .LT.0.) GO TO 26                                    ELNT 120
 25   CF=(FN/RT1)*ATAN(XP)                                              ELNT 121
      GOTO30                                                            ELNT 122
 26   IF(ABS(XP).GE.0.1)GOTO27                                          ELNT 123
      YX=XP**2                                                          ELNT 124
      YX=2.*XP*(1.+YX*(1./3.+YX*(.2+YX/7.)))                            ELNT 125
      GOTO28                                                            ELNT 126
 27   YX=ALOG((1.+XP)/(1.-XP))                                          ELNT 127
 28   CF=(FN*YX)/(2.*RT1)                                               ELNT 128
      GOTO30                                                            ELNT 129
 29   CF=0.                                                             ELNT 130
 30   BB=-1.                                                            ELNT 131
      S=SQRT(SST)                                                       ELNT 132
      C=SQRT(1.-SST)                                                    ELNT 133
      GOTO32                                                            ELNT 134
 31   SUM=2.*SUM-CF                                                     ELNT 135
      GOTO45                                                            ELNT 136
 32   U=S/C                                                             ELNT 137
      V=1./C                                                            ELNT 138
      T=U*V                                                             ELNT 139
      W=U**2                                                            ELNT 140
      IF(S.GT.0.1)GOTO33                                                ELNT 141
      R=S*(1.+S2*(1./3.+S2*(.2+S2/7.)))                                 ELNT 142
      GOTO34                                                            ELNT 143
   33 R=ALOG(U+V)                                                       ELNT 144
   34 D=1.+FN                                                           ELNT 145
      IF(D.GT.SKP) GOTO37                                               ELNT 146
C     POWER SERIFS IN 1+N AND 1 - (K SQUARED)                           ELNT 147
   35 CA=1.                                                             ELNT 148
      CB=-0.5*SKP                                                       ELNT 149
      AL=(T+R)/2.                                                       ELNT 150
      BE=U*V**3                                                         ELNT 151
      FM=0.                                                             ELNT 152
      SUM=AL                                                            ELNT 153
      T1=SUM*1.E-8                                                      ELNT 154
   36 CA=-D*CA+CB                                                       ELNT 155
      AL=(BE-(2.*FM+1.)*AL)/(2.*(FM+2.))                                ELNT 156
      X=CA*AL                                                           ELNT 157
      SUM=SUM+X                                                         ELNT 158
      IF(ABS(X).LT.T1)GOTO44                                            ELNT 159
      FM=FM+1.                                                          ELNT 160
      CB=-((2.*FM+1.)/(2.*(FM+1.)))*CB*SKP                              ELNT 161
      IF(ABS(BE).LT.10.E-30) BE=0.                                      ELNT 162
      BE=BE*W                                                           ELNT 163
      GOTO36                                                            ELNT 164
C     POWER SERIES IN 1 - (K SQUARED)                                   ELNT 165
   37 RT=SQRT(ABS(FN))                                                  ELNT 166
      IF(FN)38,39,40                                                    ELNT 167
   38 Q=ALOG((1.+RT*S)/(1.-RT*S))/(2.*RT)                               ELNT 168
      GOTO41                                                            ELNT 169
   39 Q=S                                                               ELNT 170
      GOTO41                                                            ELNT 171
   40 Q=ATAN(RT*S)/RT                                                   ELNT 172
   41 SUM=FN*Q+R                                                        ELNT 173
      PKP=SKP                                                           ELNT 174
      AP=-0.5                                                           ELNT 175
      FM=1.                                                             ELNT 176
      T1=SUM*1.E-8                                                      ELNT 177
 42   Q=(R-Q)/D                                                         ELNT 178
      R=T/(2.*FM)-(1.-.5/FM)*R                                          ELNT 179
      X=AP*(FN*Q+R)*PKP                                                 ELNT 180
      SUM=SUM+X                                                         ELNT 181
      IF(ABS(X).LT.T1)GOTO43                                            ELNT 182
      T=T*W                                                             ELNT 183
      PKP=PKP*SKP                                                       ELNT 184
      FM=FM+1.                                                          ELNT 185
      AP=-AP*(1.-.5/FM)                                                 ELNT 186
      GOTO42                                                            ELNT 187
 43   SUM=SUM/D                                                         ELNT 188
 44   IF(BB.LT.0.)GOTO31                                                ELNT 189
 45   IF(B)5,9,46                                                       ELNT 190
 46   PIF=A*SUM                                                         ELNT 191
 47   PIF=PIE+PIE*ROUND                                                 ELNT 192
      RETURN                                                            ELNT 193
C     ERROR RETURN                                                      ELNT 194
 48   PIF=0.                                                            ELNT 195
      GOTO47                                                            ELNT 196
C   CASE OF PI(1,1,PHI)                                                 ELNT 197
   50 PIE = 0.5*(TAN(P )/ COS(P )+ALOG(TAN((HP+P )/2.0)))               ELNT 198
      GO TO 47                                                          ELNT 199
      END                                                               ELNT 200
      SUBROUTINE INEL (F,E,PI,A,PHI,SKI,K3,K2,K1)
C  THIS SUBROUTINE CALCULATES THE INCOMPLETE ELLIPTIC INTEGRALS OF THE
C  FIRST, SECOND AND THIRD KINDS. THE ARGUMENTS ARE*
C        F      VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THE FIRST KIND
C        F      VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THE SECOND KIND
C        PI     VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THE THIRD KIND
C        A      VALUE OF ALPHA SQUARED
C        PHI    VALUE OF PHI
C        SKI    VALUE OF K SQUARED
C        K3     IF EQ 0, DO NOT COMPUTE PI ; IF NE 0, COMPUTE PI
C        K2     IF EQ 0, DO NOT COMPUTE E  ; IF NE 0, COMPUTE E
C        K1     IF EQ 0, DO NOT COMPUTE F  ; IF NE 0, COMPUTE F
C     DOUBLE PRECISION ARG,FD,ED
      DATA PIT/1.57079633/
      E=0.0
      F=0.0
      IF (K3.EQ.0) GO TO 220
      CALL ELINT3 (SKI,-A,PHI,PI)
  220 IF (K1.EQ.0) GO TO 240
      IF (ABS(PHI-PIT).GT.10.0**(-7)) GO TO 230
      ARG=1.0-SKI
      CALL ELLC (ARG,FD,ED,1)
      F=FD
      GO TO 240
  230 CALL ELINT3 (SKI,0.0,PHI,F)
  240 IF (K2.EQ.0) GO TO 260
      IF (ABS(PHI-PIT).GT.10.0**(-7)) GO TO 250
      ARG=1.0-SKI
      CALL ELLC (ARG,FD,ED,2)
      E=ED
      GO TO 260
  250 CALL ELINT3 (SKI,-SKI,PHI,E)
      E=(1.0-SKI)*F+0.5*SKI*SIN(2.0*PHI)/SQRT(1.0-SKI*SIN(PHI)**2)
  260 RETURN
      END
      OVERLAY(AXSY,3,0)                                                 PREP  001C
      PROGRAM PREP                                                      PREP  002C
C     SUBROUTINE  PREP                                                  PREP  003I
C **             * PREPARE TAPES 3 AND 11 FOR USE BY LINK 5 (MATSOL)    PREP  004
      COMMON/SPACER/WKAREA(5000)                                        PREP  005
      DIMENSION TEMP(105), Y2(100)                                      PREP  006
      COMMON      HEDR(10)  ,CASE       ,NB         ,NNU                PREP  007
     1            ,FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07  PREP  008
     2            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12  PREP  009
     3            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17  PREP  010
     4            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22  PREP  011
     5            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27  PREP  012
      COMMON   NT,        ND(11),   MN,        NUNA(5),  TYPEA(5),      PREP  013
     1         NER1,      NER2,     NMA,       NSIGA,    NSIGC,         PREP  014
     2         NUNC(5),   TYPEC(5), NLF(11),   IEC,      NSIGEC,        PREP  015
     3         TYPEEC(5),NUNEC(5)                                       PREP  016
C     DOUBLE PRECISION  HEDR, CASE                                      PREP  017I
      INTEGER     FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07  PREP  018
     1            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12  PREP  019
     2            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17  PREP  020
     3            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22  PREP  021
     4            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27  PREP  022
      REAL        MN                                                    PREP  023
      DIMENSION COSSQR(100), RHS(100)                                   PREP  024
      DIMENSION     A(105),    R(100,5),  FF(100),   T(100)             PREP  025
      DATA  FOURPI/12.5663706/                                          PREP  026
C***   ***AXISYMMETRIC FLOW ONLY                      MS = 0            PREP  027
C***   ***CROSS FLOW ONLY                             MS = 1            PREP  028
C***   ***EXTRA CROSS FLOW ONLY                       MS = 2            PREP  029
C***   ***AXISYMMETRIC AND CROSS FLOW                 MS = 3            PREP  030
C***   ***AXISYMMETRIC AND EXTRA CROSS FLOW           MS = 4            PREP  031
C***   ***CROSS AND EXTRA CROSS FLOW                  MS = 5            PREP  032
C***   ***AXISYMMETRIC,CROSS, AND EXTRA CROSS FLOW  MS = 6              PREP  033
      NCK1=0                                                            PREP  034
      NCK2=0                                                            PREP  035
      NCK3=0                                                            PREP  036
      NCK4=0                                                            PREP  037
      NCK5=0                                                            PREP  038
      NCK6=0                                                            PREP  039
      IF(FLG12.EQ.0.OR.(FLG04.FQ.0.AND.FLG21.EQ.0) ) GO TO 3            PREP  040
      IF (FLG05.EQ.0) GO TO 4                                           PREP  041
C***   ***SKIP OFF BODY COORDINATES                                     PREP  042
      READ(12)                                                          PREP  043
    4 NI=NT+NB                                                          PREP  044
      READ(12) (TEMP(I),I = 1,NI),(TEMP(I),I = 1,NI),                   PREP  045
     1   (TEMP(I),I = 1,NT), (Y2(I),I = 1,NT)                           PREP  046
      REWIND 12                                                         PREP  047
    3 REWIND 3                                                          PREP  048
      IF (FLG03) 5,800,5                                                PREP  049
C **            * PREPARE AXISYMMETRIC MATRIX TAPE (3)                  PREP  050
    5 IF (FLG19.GT.0) GO TO 2000                                        PREP  051
      IF ( FLG22.GT.0) GO TO 255                                        PREP  052
      K = 0                                                             PREP  053
      L = NT+NSIGA                                                      PREP  054
      READ (4) (A(I),I=1,NT),(FF(I),I=1,NT)                             PREP  055
      IF (FLG16.NE.0) GO TO 20                                          PREP  056
      K = K+1                                                           PREP  057
      DO 10 I = 1, NT                                                   PREP  058
   10 R(I,K) = A(I)                                                     PREP  059
   20 IF (NNU) 60,60,30                                                 PREP  060
   30 DO 50 J = 1, NNU                                                  PREP  061
      READ (4) MS,(A(I),I=1,NT)                                         PREP  062
      IF (MS.EQ.1.OR.MS.FQ.2.OR.MS.EQ.5) GO TO 50                       PREP  063
      K = K+1                                                           PREP  064
      DO 40 I = 1, NT                                                   PREP  065
   40 R(I,K) = A(I)                                                     PREP  066
   50 CONTINUE                                                          PREP  067
   60 IF (FLG14.LE.0) GO TO 290                                         PREP  068
      NR= NMA+1                                                         PREP  069
      READ (4) (R(I,1), I=NR,NT)                                        PREP  070
      REWIND 4
      DO 220 I = NR, NT
  220 R(I,1) = R(I,1)-FF(I)
      IF (FLG14.EQ.NB) GO TO 245
      DO 240 I = 1, NMA
      READ (9) (A(J),J=1,NT)
      A(NT+1) =  R(I,1)
  240 WRITE (3) (A(J),J=1,L)
  245 DO 250 I = NR, NT
      READ (9) (A(J),J=1,NT),(A(J),J=1,NT)
      A(NT+1) =  R(I,1)
  250 WRITE (3) (A(J),J=1,L)
C  PRESCRIBED TANGENTIAL VELOCITY      INPUT TO SOLVIT ON TAPE 3
C                                      OUTPUT FROM SOLVIT ON TAPE 3
C  TAPES 1 AND 2 ARE SCRATCH TAPES
      CALL SOLVIT(WKAREA,NT,NSIGA,5000,3,1,2,3,NCK1)
      IF(NCK1 .EQ. 1) GO TO 9010
  251 REWIND 9
      GO TO 800
C***   ***AXISYMMETRIC FLOW   *   GENERATED (RESEP) BOUNDARY CONDITIONS
C***   ***NPB1 = THE NUMBER OF ELEMENTS ON BODY 1
C***   ***NPB2 = THE NUMBER OF ELEMENTS ON BODY 2
  255 NPB1 = ND(1) - 1
      NPB2 = ND(2) - 1
      NSIGA = 3
      NSIGC = 1
      NSIGFC = 1
      L = NT + NSIGA
C***   ***L IS THE TOTAL  WIDTH OF THE MATRIX FOR AXISYMMETRIC FLOW INCL
C***   ***RIGHT HAND SIDES
      READ (4)
      READ(4) ( COSSQR(I),I = 1,NPB1), (RHS(I),I = 1,NPB1 )
      REWIND 4
      DO 260 I = 1,NPB1
      R(I,1) = 0.0
      R(I,2) = 1.0
  260 R(I,3) = COSSQR(I)
      NBEGIN = NPB1 + 1
      NEND = NPB1 + NPB2
      DO 265 I = NBEGIN,NEND
      R(I,1) = 1.0
      R(I,2) = 0.0
  265 R(I,3) = 0.0
  290 REWIND 4
      ASSIGN 400 TO M
      IF (FLG12.NE.0) ASSIGN 300 TO M
      DO 700 I = 1, NT
      GO TO M, (300,400)
  300 READ (9) (A(J),J=1,NT),(A(J),J=1,NT),(A(J),J=1,NT)
      GO TO 500
  400 READ (9) (A(J),J=1,NT)
  500 DO 600 J = 1, NSIGA
      K = NT+J
  600 A(K)=  R(I,J)
  700 WRITE (3) (A(J),J=1,L)
C  AXISYMMETRIC FLOW          INPUT TO SOLVIT ON TAPE 3
C                             OUTPUT FROM SOLVIT ON TAPE 3
C  TAPES 1 AND 2 ARE SCRATCH TAPES
      CALL SOLVIT(WKAREA,NT,NSIGA,5000,3,1,2,3,NCK2)
      IF(NCK2 .EQ. 1) GO TO 9020
  701 REWIND 9
C **           * PREPARE CROSSFLOW MATRIX TAPE (11)
C **           * SKIP SINA * READ COSA
  800 IF (FLG04.EQ.0) GO TO 1610
      K = 0
      L = NT+NSIGC
      IF (FLG22.GT.0) GO TO 910
      READ (4) (A(I),I=1,NT),(A(I),I=1,NT)
      IF (FLG17.NE.0) GO TO 820
      K = K+1
      DO 810 I = 1, NT
  810 R(I,K) = A(I)
  820 IF (NNU) 900,900,830
  830 DO 850 J = 1, NNU
      READ (4) MS,(A(I),I=1,NT)
      IF ( MS.EQ.0.OR.MS.EQ.2.OR.MS.EQ.4) GO TO 850
      K = K+1
      DO 840 I = 1, NT
  840 R(I,K) = -A(I)
  850 CONTINUE
  900 REWIND 4
      GO TO 1000
C***   ***CROSS FLOW    *    GENERATED (RESEP) BOUNDARY CONDITIONS
  910 DO 920 I = 1,NPB1
  920 R(I,1) = -RHS(I)
      DO 930 I = NBEGIN,NEND
  930 R(I,1) = 0.0
 1000 ASSIGN 1300 TO M
      IF (FLG12.NE.0) ASSIGN 1200 TO M
      DO 1600 I = 1, NT
      GO TO M, (1200,1300)
 1200 READ (10) (A(J),J=1,NT),(A(J),J=1,NT),(A(J),J=1,NT)
C***   ***FORM PHI MATRIX FROM THETA (CROSS FLOW) MATRIX
      DO 1250 J = 1,NT
 1250 A(J) = Y2(I) * A(J)
      GO TO 1400
 1300 READ (10) (A(J),J=1,NT)
 1400 DO 1500 J = 1, NSIGC
      K = NT+J
 1500 A(K) =-R(I,J)
 1600 WRITE (11) (A(J),J=1,L)
C  CROSS FLOW        INPUT TO SOLVIT ON TAPE 11
C                    OUTPUT FROM SOLVIT ON TAPF 3
C  TAPES 1 AND 2 ARE SCRATCH TAPES
      CALL SOLVIT(WKAREA,NT,NSIGC,5000,11,1,2,3,NCK3)
      IF(NCK3 .EQ. 1) GO TO 9030
 1605 REWIND 10
 1610 CONTINUE
C***   ***FXTRA CROSS FLOW
      REWIND 11
      IF (FLG21.EQ.0.AND.FLG22.EQ.0)RETURN
      K = 0
      L = NT + NSIGEC
      IF (FLG22.GT.0) GO TO 1800
C***   ***EXTRA CROSS FLOW   *   NON-UNIFORM FLOW ONLY
C***   ***SKIP RECORD WITH STNES AND COSINES
      READ (4)
      DO 1650 J=1,NNU
      READ(4) MS, (A(I),I=1,NT)
      IF (MS.LT.2.OR.MS.EQ.3) GO TO 1650
      K = K+ 1
      DO 1640 I = 1,NT
 1640 R(I,K) = A(I)
 1650 CONTINUE
      GO TO 1900
C***   ***EXTRA CROSS FLOW   *   GENERATED (RESEP) BOUNDARY CONDITIONS
 1800 DO 1820 I = 1,NPB1
 1820 R(T,1) = COSSQR(I)
      DO 1840 I = NBEGIN,NEND
 1840 R(I,1) = 0.0
 1900 REWIND 4
C***   ***M IS 1920   *   SOLVE A MATRIX
      ASSIGN 1920 TO M
C***   ***M IS 1940   *   SOLVE POTENTIAL MATRIX
      IF (FLG12.NE.0)ASSIGN 1940 TO M
      DO 1980 I = 1,NT
      GO TO M, (1920,1940)
C***   ***SOLVF A MATRIX
 1920 READ (8) (A(J),J =1,NT)
      GO TO 1960
 1940 READ (8) (A(J),J=1,NT),(A(J),J=1,NT),(A(J),J=1,NT)                 PREP  211
C***   ***FORM PHI MATRIX FROM THETA (EXTRA CROSS FLOW) MATRIX             PREP  212
      DO 1950 J = 1,NT                                                     PREP  213
 1950 A(J) = Y2(I) * A(J) / 2.0                                            PREP  214
 1960 DO 1970 J = 1,NSIGEC                                                 PREP  215
      K = NT + J                                                           PREP  216
 1970 A(K) = R(I,J)                                                        PREP  217
 1980 WRITE (11) (A(J),J=1,L)                                              PREP  218
C***   ***EXTRA CROSS FLOW   INPUT TO SOLVIT ON TAPE 11                    PREP  219
C***   ***OUTPUT FROM SOLVIT ON TAPE 3                                     PREP  220
C***   ***TAPES 1 AND 2 ARE SCRATCH TAPES                                  PREP  221
      CALL SOLVIT (WKAREA,NT,NSIGEC,5000,11,1,2,3,NCK4)                    PREP  222
      IF(NCK4 .EQ. 1) GO TO 9040                                           PREP  223
 1985 REWIND 8                                                             PREP  224
      REWIND 11                                                            PREP  225
C     RETURN                                                               PREP  226I
      GO TO 9070                                                           PREP  227C
 2000 IF(FLG23 .GT. 0)GO TO 3000                                           PREP  228
      NR = NT - NMA                                                        PREP  229
      L = NMA+1                                                            PREP  230
      READ (4) (R(I,1),I=1,NMA)                                            PREP  231
      READ (4) (FF(I),I=1,NR)                                              PREP  232
      DO 2100 I = 1, NR                                                    PREP  233
 2100 FF(I) = FF(I)/FOURPI                                                 PREP  234
      BACKSPACE 4                                                          PREP  235
      WRITE (4) (FF(I),I=1,NR)                                             PREP  236
      REWIND 4                                                             PREP  237
      DO 2300 I = 1, NMA                                                   PREP  238
      READ (9) (A(J),J=1,NMA),(T(J),J=1,NR)                                PREP  239
      DO 2200 J = 1, NR                                                    PREP  240
 2200 R(I,1) = R(I,1)- T(J)*FF(J)                                          PREP  241
      A(L) =  R(I,1)                                                       PREP  242
 2300 WRITE (3) (A(J),J=1,L)                                               PREP  243
C  PRESCRIBED VORTICITY      INPUT FOR SOLVIT ON TAPE 3                    PREP  244
C                            OUTPUT FROM SOLVIT ON TAPE 3                  PREP  245
```

```
C  TAPFS 1 AND 2 ARE SCRATCH TAPES
      CALL SOLVIT(WKAREA,NMA,L - NMA,5000,3,1,2,3,NCK5)
      IF(NCK5 .EQ. 1) GO TO 9000
 2500 REWIND 9
      GO TO 800
 3000 NR = NT - NMA
      NMAP1 = NMA + 1
C*** *  CALCULATE THE NUMBER OF RHS
C
      LL = 0
      DO 3100 I=1,NB
      IF( NLF(I) .GT. 0 )GO TO 3100
      LL = LL + 2
 3100 CONTINUE
      L = NMAP1 + LL
C
C*** *  READ SINS FOR STREAMFLOW RHS
C
      READ(4)( R(I,1),I=1,NMA )
C
C*** *  READ  INPUT PRESCRIBED VORTICITIES
C
      READ(4)( FF(I),I=NMAP1,NT )
      WRITE(6,8001) (FF(I),I=NMAP1,NT)
 8001 FORMAT(1H1,* THE INPUT PV ARE */(6E20.7))
      DO 3125 I = NMAP1,NT
 3125 FF(I) = FF(I) / (-FOURPI)
C
C*** *  READ STRIP VORTEX RHS
C
      LLD2P1 = LL/2 + 1
      DO 3150 J=2,LLD2P1
 3150 READ(4)MS,( R(I,J),I=1,NMA )
C
C*** *  IPV IS BODY NUMBER OF 1ST PRESCRIBED VORTICITY BODY
```

```
C
      IPV = NB - FLG14 + 1
      JBOD = LLD2P1
      NN = NMA
C
      DO 3300 KCNT = IPV,NB
      JBOD = JBOD + 1
      NN = NN + ND(KCNT) - 1
C
C*** *  READ COLUMN OF RHS CALCULATED BY NOTS FORMULA
C
      READ(4) (R(ICNT,JBOD),ICNT = 1,NMA)
C
C*** *  MULTIPLY NOTS COLUMN BY LAST PRESCRIBED VORTICITY ON THAT BODY
C
      DO 3300 ICNT =1,NMA
 3300 R(ICNT,JBOD) = R(ICNT,JBOD) * FF(NN) *(-FOURPI)
C
      REWIND 4
      DO 3400 I=1,NMA
      NEND = NMA
      JBOD = LLD2P1
      READ(9) ( A(J),J=1,NMA),( T(J),J=NMAP1,NT )
C
      DO 3350 KCNT=IPV,NB
      JBOD = JBOD + 1
      NBEG = NEND + 1
      NEND = NEND + ND(KCNT) -1
C
C*** *  SUM PV VORTEX ELEMENTS * INPUT PV AND ADD TO NOTS RHS
C
      DO 3350 NCNT=NBEG,NEND
C
C   ***   WHEN COMING OFF UNIT 9, THE VORTEX ELEMENTS( T(J) ) ARE STILL
C   ***   WHILE NOTS COLUMNS COMING OFF UNIT 4 ARE RHS.   THE TWO SHOULD
```

```
C   ***   ADDED TO FORM A COMPLETE RHS, BUT THEY ARE SUBTRACTED SINCE TH
C   ***   OF T(J)  MUST BE CHANGED
C
 3350 R(I,JBOD) = R(I,JBOD) - T(NCNT) * FF(NCNT)
C
C*** *  ATTACH ALL RHS FOR ROW NUMBER I
C
      LRHS = 0
      DO 3375  ICNT=NMAP1,L
      LRHS = LRHS + 1
 3375 A(ICNT) = R(I,LRHS)
C
 3400 WRITE(3)(A(J),J=1,L)
C*** *  RING WING OPTION          INPUT FOR SOLVIT ON TAPE 3
C*** *                            OUTPUT FOR SOLVIT ON TAPE 3
C
      CALL SOLVIT(WKAREA,NMA,L-NMA,5000,3,1,2,3,NCK6)
      IF(NCK6 .EQ. 1) GO TO 9050
 3500 REWIND 9
      GO TO 800
 9000 WRITE(6,9001)
 9001 FORMAT(61H NOT ENOUGH SPACE RESERVED IN SOLVIT FOR PRESCRIBED VORT
     1ICITY)
      GO TO 9080
 9010 WRITE(6,9011)
 9011 FORMAT(71H NOT ENOUGH SPACE RESERVED IN SOLVIT FOR PRESCRIBED TANG
     1ENTIAL VELOCITY)
      GO TO 9080
 9020 WRITE (6,9021)
 9021 FORMAT(58H NOT ENOUGH SPACE RESERVED IN SOLVIT FOR AXISYMMETRIC FL
     1OW)
      GO TO 9080
 9030 WRITE(6,9031)
 9031 FORMAT(51H NOT ENOUGH SPACE RESERVED IN SOLVIT FOR CROSS FLOW)
      GO TO 9080
```

```
9040 WRITE (6,9041)                                                    PREP  351
9041 FORMAT (57H NOT ENOUGH SPACE RESERVED TN SOLVIT FOR EXTRA CROSS FL PREP  352
    1OW)                                                               PREP  353
     GO TO 9080                                                        PREP  354
9050 WRITE(6,9051)                                                     PREP  355
9051 FORMAT(51H NOT ENOUGH SPACF RESERVED IN SOLVIT FOR RING WING )    PREP  356
9070 CONTINUE                                                          PREP  357C
9080 STOP                                                              PREP  358
     END                                                               PREP  359
```

```
      SUBROUTINE  SOLVIT (A, ND, MD, KD, NI, MM, NO, NW, NCK)
C
C
C           ****    ***/  *******  *****  *       ***/          *
C           *   *  *  /*     *     *      *      *  /*        ******
C     ****  ****   * / *     *     ***    *      * / *  ****     *  *
C           *      */  *     *     *      *      */  *           ***
C           *      /***      *     *      *****  /***            *  *
C
C             D I R E C T   M A T R I X   S O L U T I O N
C
C     WRITTEN BY J. L. HESS * PROGRAMMED BY T. M. RIDDELL
C
      DIMENSION  A ( KD )
C
      LOGICAL  LAST
C
C     CALL TIMEV(AA1)
      NCK=0
      N = ND
      M = MD
      KORE = KD
      NPM = N + M
      IF (MAXO(3 * NPM, M * N) .GT. KORE)  NCK= 1
      MT = MM
      REWIND MT
      NIN = NI
      REWIND NIN
      NOUT = NO
      REWIND NOUT
      MP1 = M + 1
      NN = N
      NEL = NPM
C
C - - CALCULATE THE MAXIMUM NO. OF ROWS, *K*
```

```
C                                                                         SOLV  036
   10 K = (KORE - NEL) / NEL                                              SOLV  037
C                                                                         SOLV  038
C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE              SOLV  039
C                                                                         SOLV  040
      LAST = K .GE. NN                                                    SOLV  041
      IF (LAST) K = NN                                                    SOLV  042
C                                                                         SOLV  043
C - - READ ≠K≠ ROWS OF THE AUGMENTED ≠A≠ MATRIX                           SOLV  044
C                                                                         SOLV  045
   30 NT = 0                                                              SOLV  046
      DO 40 IB = 1, K                                                     SOLV  047
      NS = NT + 1                                                         SOLV  048
      NT = NT + NEL                                                       SOLV  049
   40 READ (NIN) (A(IO), IO = NS, NT)                                     SOLV  050
C                                                                         SOLV  051
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROW  SOLV  052
C                                                                         SOLV  053
      IF (K .EQ. 1) GO TO 90                                              SOLV  054
C                                                                         SOLV  055
C - - ≠K≠ IS GREATER THAN ≠1≠ SO WE CAN START THE TRIANGULARIZATION       SOLV  056
C                                                                         SOLV  057
      NELP1 = NEL + 1                                                     SOLV  058
      NS = - NEL                                                          SOLV  059
      NELP2 = NELP1 + 1                                                   SOLV  060
C                                                                         SOLV  061
C - - FORM THE ≠TRAPEZOIDAL≠ ARRAY  (8)                                   SOLV  062
C                                                                         SOLV  063
      DO 50 IB = 2, K                                                     SOLV  064
      NP = NELP2 - IB                                                     SOLV  065
      NS = NS + NELP1                                                     SOLV  066
      NT = NS                                                             SOLV  067
      DO 50 IO = IB, K                                                    SOLV  068
      NT = NT + NEL                                                       SOLV  069
      MN = NT                                                             SOLV  070
      NB = NS
      A(NT) = (-A(NT)) / A(NS)
      DO 50 NF = 2, NP
      MN = MN + 1
      NB = NB + 1
   50 A(MN) = A(MN) + A(NT) * A(NB)
      IF (LAST) GO TO 90
C
C - - WRITE THE ≠TRAPEZOIDAL≠ MATRIX ON TAPE
C
      NT = 0
      NP = NFL
      NS = - NEL
      DO 60 IO = 1, K
      NS = NS + NELP1
      NT = NT + NEL
      WRITE (MT) NP, (A(IB), IB = NS, NT)
   60 NP = NP - 1
      NP = NP - M
      NS = KORE - NEL + 1
C
C - - READ ANOTHER ROW
C
      DO 80 IO = 1, NP
      READ (NIN) (A(IB), IB = NS, KORE)
C
C - - MODIFY THIS ROW BY THE ≠TRAPEZOIDAL≠ ARRAY
C
      NT = 1
      MN = NS
      DO 70 IB = 1, K
      NB = NT
      NF = MN + 1
      A(MN) = (-A(MN)) / A(NT)
      DO 65 NN = NF, KORF
      NB = NB + 1
   65 A(NN) = A(NN) + A(MN) * A(NB)
      MN = NF
   70 NT = NT + NELP1
C
C - - WRITF THE MODIFIED ROW ON TAPE
C
   80 WRITF (NOUT)      (A(NT), NT = MN, KURE)
      REWIND NOUT
      REWIND NIN
C
C - - SWITCH THE TAPFS
C
      NT = NIN
      NIN = NOUT
      NOUT = NT
C
C - - RE-CALCULATE ROW LENGTH AND LOOP BACK
C
      NEL = NEL - K
      NN = NEL - M
      GO TO 10
C
C - - REWIND ALL TAPES
C
   90 REWIND MT
      REWIND NIN
      REWIND NOUT
C
C - - CONDFNSE THE MATRIX
C
      NN = NEL
      NL = NFL + 1
      IF (K .EQ. 1) GO TO 105
      NS = 1
      NT = NFL
      DO 100 IB = 2, K
      NS = NS + NELP1
      NT = NT + NEL
      DO 100 IO = NS, NT
      A(NL) = A(IO)
  100 NL = NL + 1
  105 N1 = KORF - K * M + 1
C
C - - THERE, NOW WE CAN START THE BACK-SOLUTION
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(N1)
C
      NREM = N
      NEL = NPM
      LAST = K .EQ. N
      NPASS = 0
C
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO *K* ROWS
C
  110 KM1 = K - 1
      KP1 = K + 1
      NS = NL - MP1
      NPASS = NPASS + 1
      DO 130 MN = 1, M
      NF = NS + MN
      A(NF) = A(NF) / A(NS)
      NT = NS
      IF (KM1 .EQ. 0) GO  TO 130
      DO 125 IB = 1, KM1
      NF = NF - IB - M
      NT = NT - MP1 - IB
      SUM = 0.0
      NP = NF
      N2 = MP1 + IB
      DO 120 IO = 1, IB
      NN = NT + IO
      NP = NP + N2 - IO
  120 SUM = SUM + A(NN) * A(NP)
  125 A(NF) = (A(NF) - SUM) / A(NT)
  130 CONTINUE
C
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(N1)
C
      N1 = KORE + 1
      DO 140 NN = 1, K
      DO 135 MN = 1, M
      NL = NL - 1
      N1 = N1 - 1
  135 A(N1) = A(NL)
  140 NL = NL - NN
C
C - - WRITE THE SOLUTIONS ON TAPE
C
      WRITE (NIN) K
      NS = N1 - 1
      DO 145 MN = 1, M
      NT = NS + MN
  145 WRITE ( NIN ) (A(IO), IO = NT, KORE, M)
C
C - - TEST IF THIS IS THE LAST PASS
C
      IF (LAST) GO TO 200
C
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF
C     THE SOLUTIONS OBTAINED SO FAR  (EQ 21)
C * * NOTE..LOCATIONS A(1) TO A(N1-1) ARE NOW FREE TO USE
C
C - - CALCULATE THE NEXT VALUES OF ≠NEL≠ AND ≠NREM≠
C
      NELOLD = NEL
      KOLD = K                                                          SOLV 211
      NEL = NEL - K                                                     SOLV 212
      NREM = NREM - K                                                   SOLV 213
C                                                                       SOLV 214
C - - NOW APPLY THE INCREDIBLE FORMULA FOR THE NEW *K*                  SOLV 215
C                                                                       SOLV 216
      K = (-4 * M - 1) / 2 + IFIX(SQRT(0.25 + FLOAT((4 * M + 2) * M +   SOLV 217
     1 2 * (KORE - NELOLD))))                                           SOLV 218
      NROW = NREM - K + 1                                               SOLV 219
      IF (K .LT. NREM) GO TO 150                                        SOLV 220
      LAST = .TRUE.                                                     SOLV 221
      NROW = 1                                                          SOLV 222
      K = NREM                                                          SOLV 223
  150 NS = 1                                                            SOLV 224
      NT = NELOLD + 1                                                   SOLV 225
C                                                                       SOLV 226
C - - READ IN THE ROWS TO BE MODIFIED                                   SOLV 227
C                                                                       SOLV 228
      DO 190 IB = 1, NREM                                               SOLV 229
      NT = NT - 1                                                       SOLV 230
      IF (IB .LE. NROW) GO TO 160                                       SOLV 231
      NS = NS + NN                                                      SOLV 232
      NT = NT + NN                                                      SOLV 233
  160 READ ( MT ) NN, (A(IO), IO = NS, NT)                              SOLV 234
      NP = N1 - 1                                                       SOLV 235
      NF = NT - M - KM1                                                 SOLV 236
      NN = NN - KOLD                                                    SOLV 237
      DO 170 MN = 1, M                                                  SOLV 238
      N2 = NF                                                           SOLV 239
      NA = NP + MN                                                      SOLV 240
      NB = NA                                                           SOLV 241
      SUM = 0.0                                                         SOLV 242
      DO 165 IO = 1, KOLD                                               SOLV 243
      SUM = SUM + A(N2) * A(NA)                                         SOLV 244
      N2 = N2 + 1                                                       SOLV 245
  165 NA = NA + M                                                       SOLV  246
      N2 = N2 + MN - 1                                                  SOLV  247
  170 A(N2) = A(N2) - SUM                                               SOLV  248
C                                                                       SOLV  249
C - - WRITF THE MODIFIED ROW ON TAPE OR CUNDFNSE THE ROW                 SOLV  250
C                                                                       SOLV  251
      NL = NT - M + 1                                                   SOLV  252
      IF (IB .GE. NROW) GO TO 175                                       SOLV  253
      NF = NL - KP1                                                     SOLV  254
      WRITF (NOUT) NN, (A(IO), IO = NS, NF), (A(IO), IO = NL, NT)       SOLV  255
      GO TO 190                                                         SOLV  256
  175 NF = NL - KOLD                                                    SOLV  257
      DO 180 MN = NL, NT                                                SOLV  258
      A(NF) = A(MN)                                                     SOLV  259
  180 NF = NF + 1                                                       SOLV  260
  190 CONTINUE                                                          SOLV  261
      REWIND MT                                                         SOLV  262
      REWIND NOUT                                                       SOLV  263
C                                                                       SOLV  264
C - - SWITCH THE TAPES                                                  SOLV  265
C                                                                       SOLV  266
      NT = MT                                                           SOLV  267
      MT = NOUT                                                         SOLV  268
      NOUT = NT                                                         SOLV  269
C                                                                       SOLV  270
C - - LOOP BACK THRU THE SOLUTION                                       SOLV  271
C                                                                       SOLV  272
      NL = NF                                                           SOLV  273
      GO TO 110                                                         SOLV  274
C                                                                       SOLV  275
C - - START TO WRAP IT UP                                               SOLV  276
C                                                                       SOLV  277
  200 REWIND NIN                                                        SOLV  278
      N2 = N                                                            SOLV  279
C                                                                       SOLV  280
```

```
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KORE) ARE FREE       SOLV  281
C                                                                        SOLV  282
      DO 220 IB = 1, NPASS                                               SOLV  283
      READ (NIN) K                                                       SOLV  284
      N1 = N2 - K + 1                                                    SOLV  285
      NS = N1                                                            SOLV  286
      NT = N2                                                            SOLV  287
C                                                                        SOLV  288
C - - READ IN THE SOLUTIONS                                              SOLV  289
C                                                                        SOLV  290
      DO 210 IO = 1, M                                                   SOLV  291
      READ (NIN) (A(NN), NN = NS, NT)                                    SOLV  292
      NT = NT + N                                                        SOLV  293
  210 NS = NS + N                                                        SOLV  294
  220 N2 = N1 - 1                                                        SOLV  295
C                                                                        SOLV  296
C  ---  REWIND ALL INPUT TAPES                                           SOLV  297
      REWIND NIN                                                         SOLV  298
      REWIND MT                                                          SOLV  299
      REWIND NOUT                                                        SOLV  300
C - - WRITE THE SOLUTIONS ON TAPE                                        SOLV  301
C                                                                        SOLV  302
      NT = 0                                                             SOLV  303
      DO 230 IO = 1, M                                                   SOLV  304
      NS = NT + 1                                                        SOLV  305
      NT = NT + N                                                        SOLV  306
  230 WRITE (NW) (A(NN), NN = NS, NT)                                    SOLV  307
C                                                                        SOLV  308
C     CALL TIMEV(AA2)                                                    SOLV  309I
C     BB = (AA2 - AA1) / 60.                                             SOLV  310I
C     WRITE (6, 300) N, N, M, BB                                         SOLV  311I
C 300 FORMAT (4HOTHE I5, 2H X I5, 12H MATRIX WITH I4, 35H RIGHT SIDES WA SOLV  312I
C    1S SOLVED DIRECTLY IN F8.3, 9H MINUTES. )                           SOLV  313I
      RETURN                                                             SOLV  314
      END                                                                SOLV  315
      OVERLAY(AXSY,4,0)                                                  PAR4  001C
      PROGRAM PART4                                                      PAR4  002C
C     SUBROUTINE PART4                                                   PAR4  003I
C                                                                        PAR4  004
C              * COMPUTE VELOCITY COMPONENTS AND PRINT                   PAR4  005
C                                                                        PAR4  006
      COMMON /IPSF/ PSF                                                  PAR4  007
      COMMON       HEDR(10)   ,CASE       ,NB         ,NNU               PAR4  008
     1             ,FLG03     ,FLG04      ,FLG05      ,FLG06     ,FLG07  PAR4  009
     2             ,FLG08     ,FLG09      ,FLG10      ,FLG11     ,FLG12  PAR4  010
     3             ,FLG13     ,FLG14      ,FLG15      ,FLG16     ,FLG17  PAR4  011
     4             ,FLG18     ,FLG19      ,FLG20      ,FLG21     ,FLG22  PAR4  012
     5             ,FLG23     ,FLG24      ,FLG25      ,FLG26     ,FLG27  PAR4  013
      COMMON    NT,        ND(11),    MN,        NUNA(5),  TYPEA(5),     PAR4  014
     1          NER1,      NER2,      NMA,       NSIGA,    NSIGC,        PAR4  015
     2          NUNC(5),   TYPEC(5),  NLF(11),   IEC,      NSIGEC,       PAR4  016
     3          TYPEEC(5),NUNEC(5)                                       PAR4  017
C     DOUBLE PRECISION  HEDR, CASE                                       PAR4  018I
      COMMON /COMBIN/CHAY(2)                                             PAR4  019
      INTEGER      FLG03      ,FLG04      ,FLG05      ,FLG06     ,FLG07  PAR4  020
     1             ,FLG08     ,FLG09      ,FLG10      ,FLG11     ,FLG12  PAR4  021
     2             ,FLG13     ,FLG14      ,FLG15      ,FLG16     ,FLG17  PAR4  022
     3             ,FLG18     ,FLG19      ,FLG20      ,FLG21     ,FLG22  PAR4  023
     4             ,FLG23     ,FLG24      ,FLG25      ,FLG26     ,FLG27  PAR4  024
      REAL         MN                                                    PAR4  025
C                                                                        PAR4  026
      COMMON /C4/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),   PAR4  027
     1              SINA(100),CUSA(100),XP(100),  YP(100)                PAR4  028
      COMMON /TC/   RB(100,10),     SIG(100,5),   A(100),   B(100),      PAR4  029
     1              Z(100),         PHI(100,5),   XN(100,5),T(100,5),    PAR4  030
     2              T3(100,5),      NSIG,         NP,       NI,          PAR4  031
     3              SUMV,           SUMM(5)                              PAR4  032
C                                                                        PAR4  033
C              * START                                                   PAR4  034
      DO 20 J = 1,10                                                     PAR4  035
      DO 20 I = 1,500
   20 RB(I,J) = 0.0
      REWIND 3
      IF (FLG05.EQ.0) GO TO 30
C                * READ OFF-BODY XP,YP
      NP=ND(NB+1)
      READ (12) (XP(I),I=1,NP),(YP(I),I=1,NP)
C                * READ X1,Y1,X2,Y2,DELS WITH MACH NO. ADJUSTMENT IF ANY
   30 NI=NT+NB
      READ (12)   (X1(I),I=1,NI),(Y1(I),I=1,NT),(X2(I),I=1,NT)
     1           ,(Y2(I),I=1,NT),(DELS(I),I=1,NT)
C                * READ SINA,COSA,NO,TO..
      READ (4) (A(I),I=1,NT),(B(I),I=1,NT)
      NMAP1 = NMA + 1
      IF(FLG23 .GT. 0)READ(4)(Z(I),I=NMAP1,NT)
      SUMV = 0.0
      DO 100 I = 1, NT
      SINA(I) = A(I)
      COSA(I) = B(I)
  100 SUMV = SUMV + B(I)*DELS(I)*Y2(I)**2
      SUMV = SUMV*3.141593
      IF (FLG03.LE.0) GO TO 1000
      L = 1
      LS = 0
      IF (FLG16.NE.0) GO TO 200
      DO 150 I = 1, NT
      RB(I,L) = A(I)
  150 RB(I,L+1) = B(I)
  200 IF (NNU) 600,600,300
  300 DO 500 J = 1, NNU
      READ (4) MS,(A(I),I=1,NT),(B(I),I=1,NT)
      IF (MS.EQ.1.OR.MS.EQ.2.OR.MS.EQ.5) GO TO 500
      L = L+2
      LS = LS+1
      IF (LS.EQ.1.AND.FLG16.GT.0) L=L-2
      DO 400 I = 1, NT
      RB(I,L) = A(I)
  400 RB(I,L+1) = B(I)
  500 CONTINUE
      IF(FLG23 .LE. 0)GO TO 600
      IPV = NB - FLG14 + 1
      NN = NMA
      DO 550 KCNT = IPV,NB
      L = L + 2
      NN = NN + ND(KCNT) - 1
C    ***   READ NOTS COLUMNS OF RHS
      READ(4)(RB(I,L),I=1,NT),(RB(I,L+1),I=1,NT)
C    ***   MULTIPLY NOTS COLUMN BY LAST INPUT PV ON THAT BODY
      DO 550 I=1,NT
      RB(I,L) = RB(I,L) * Z(NN)
  550 RB(I,L+1) = RB(I,L+1) * Z(NN)
  600 REWIND 4
      NSIG = NSIGA
      IF(FLG23 .GT. 0)NSIG = 2.0 * NSIG - 1
C     CALL AXIS
      CALL OVERLAY (4HAXSY,4,1,6HRECALL )
 1000 IF (FLG04.LE.0) GO TO 2000
      IF (FLG03.LE.0) GO TO 1050
      READ (4) (A(I),I=1,NT),(B(I),I=1,NT)
 1050 L = 1
      LS=0
      IF (FLG17.NE.0) GO TO 1200
      DO 1100 I = 1, NT
      RB(I,L) = A(I)
 1100 RB(I,L+1) = B(I)
 1200 IF (NNU) 1600,1600,1300
 1300 DO 1500 J = 1, NNU
      READ (4) MS,(A(I),I=1,NT),(B(I),I=1,NT)
      IF ( MS.EQ.0.OR.MS.EQ.2.OR.MS.FQ.4) GO TO 1500
      L = L+2
      LS=LS+1                                                           PAR4  106
      IF (LS.EQ.1.AND.FLG17.GT.0) L=L-2                                 PAR4  107
      DO 1400 I = 1, NT                                                 PAR4  108
      RB(I,L) = A(I)                                                    PAR4  109
 1400 RB(I,L+1) = B(I)                                                  PAR4  110
 1500 CONTINUE                                                          PAR4  111
 1600 REWIND 4                                                          PAR4  112
      NSIG = NSIGC                                                      PAR4  113
C     CALL CROSS                                                        PAR4  114I
      CALL OVERLAY (4HAXSY,4,2,6HRECALL )                               PAR4  115C
C2000 IF (FLG21.LE.0) RETURN                                            PAR4  116I
 2000 IF (FLG21.LE.0)  GO TO 2500                                       PAR4  117C
 2050 REWIND 4                                                          PAR4  118
      IF(FLG22.GT.0) GO TO 2400                                         PAR4  119
      L = 0                                                             PAR4  120
C***   ***IF CONTROL REACHES THIS POINT, THERE IS AT LEAST 1 NNU        PAR4  121
C***   ***SKIP RECORD WITH SIN AND COS                                  PAR4  122
      READ (4)                                                          PAR4  123
      DO 2200 J = 1,NNU                                                 PAR4  124
      READ(4) MS, ( A(I),I=1,NT ), ( B(I),I=1,NT )                      PAR4  125
      L = L + 1                                                         PAR4  126
      DO 2200 I = 1,NT                                                  PAR4  127
      RB(I,L) = A(I)                                                    PAR4  128
 2200 RB(I,L+1) = B(I)                                                  PAR4  129
 2400 REWIND 4                                                          PAR4  130
      NSIG = NSIGEC                                                     PAR4  131
C***   ***CALL TO EXCROS FOR GENERATED (RESEP) BOUNDARY CONDITIONS      PAR4  132
C     CALL EXCROS                                                       PAR4  133I
      CALL OVERLAY (4HAXSY,4,3,6HRECALL )                               PAR4  134C
C     RETURN                                                            PAR4  135I
 2500 CONTINUE                                                          PAR4  136C
      END                                                               PAR4  137
      OVERLAY(AXSY,4,1)                                                          AXIS 001C
      PROGRAM AXIS                                                               AXIS 002C
C     SUBROUTINE AXIS                                                            AXIS 003I
C                                                                                AXIS 004
C              * COMPUTE AXISYMMETRIC VELOCITY COMPONENTS AND PRINT              AXIS 005
C                                                                                AXIS 006
      COMMON /COMBIN/CHAY(2)                                                     AXIS 007
      COMMON /IPSF/ PSF                                                          AXIS 008
      COMMON       HEDR(10)  ,CASE       ,NB         ,NNU                        AXIS 009
     1            ,FLG03     ,FLG04      ,FLG05      ,FLG06      ,FLG07          AXIS 010
     2            ,FLG08     ,FLG09      ,FLG10      ,FLG11      ,FLG12          AXIS 011
     3            ,FLG13     ,FLG14      ,FLG15      ,FLG16      ,FLG17          AXIS 012
     4            ,FLG18     ,FLG19      ,FLG20      ,FLG21      ,FLG22          AXIS 013
     5            ,FLG23     ,FLG24      ,FLG25      ,FLG26      ,FLG27          AXIS 014
      COMMON    NT,        ND(11),    MN,        NUNA(5),  TYPEA(5),             AXIS 015
     1          NER1,      NER2,      NMA,       NSIGA,    NSIGC,                AXIS 016
     2          NUNC(5),   TYPEC(5),  NLF(11),   IFC,      NSIGEC,               AXIS 017
     3          TYPFEC(5),NUNEC(5)                                               AXIS 018
C     DOUBLE PRECISION  HEDR, CASE                                               AXIS 019I
      INTFGER      FLG03     ,FLG04      ,FLG05      ,FLG06      ,FLG07          AXIS 020
     1            ,FLG08     ,FLG09      ,FLG10      ,FLG11      ,FLG12          AXIS 021
     2            ,FLG13     ,FLG14      ,FLG15      ,FLG16      ,FLG17          AXIS 022
     3            ,FLG18     ,FLG19      ,FLG20      ,FLG21      ,FLG22          AXIS 023
     4            ,FLG23     ,FLG24      ,FLG25      ,FLG26      ,FLG27          AXIS 024
      REAL         MN                                                            AXIS 025
C                                                                                AXIS 026
      COMMON /C4/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),           AXIS 027
     1              SINA(100),COSA(100),XP(100),  YP(100)                        AXIS 028
      COMMON /TC/   RB(100,10),    SIG(100,5),    A(100),   B(100),              AXIS 029
     1              Z(100),        PHI(100,5),    XN(100,5),T(100,5),            AXIS 030
     2              T3(100,5),     NSIG,          NP,       NI,                  AXIS 031
     3              SUMV,          SUMM(5)                                       AXIS 032
      COMMON/ITERF/ ITER                                                         AXIS 033
C                                                                                AXIS 034
      DIMENSION UB(100),YB(100),XB(100)                                          AXIS 035
      DIMENSION     VX(100,5),      VY(100,5),      VT(100,5),
     1              TH(100,5),      CP(100,5),      SUMTDS(5)
C
      DATA FOURPI /12.5665706/
      EQUIVALENCE  ( VX(1,1) , XN(1,1) ) , (VY(1,1),T(1,1) ) ,
     1  ( VT(1,1), T3(1,1) ), ( TH(1,1),SIG(1,1) ), (CP(1,1),T3(1,1) )
C
C                * START
      NC=NT
      IF (FLG19.GT.0) NC=NMA
      IF (FLG08.EQ.0) GO TO 10
C                * TITLE FOR MATRIX PRINT
      WRITE(6,150)HEDR,CASE,PSF
      WRITE (6,8)
    8 FORMAT (1H 43H MATRICES A,B,Z BY ROWS * AXISYMMETRIC FLOW //)
C                * READ AXIS SIGMAS
   10 DO 20 N=1,NSIG
      SUMM(N)=0.0
      SUMTDS(N)=0.0
   20 READ (3) (SIG(I,N),I=1,NC)
      IF (FLG19.LE.0) GO TO 25
      READ (4)
      NR = NMA+1
      IF(FLG23 .GT. 0)GO TO 21
      READ (4) (SIG(I,1),I=NR,NT)
      REWIND 4
      GO TO 25
C
C
C*** *  RING WING
C
   21 LIFBOD = 0
      DO 22 K = 1,NB
      IF( NLF(K) .GT. 0)GO TO 22
      LIFBOD = LIFBOD + 1
   22 CONTINUF
      LBP1 = LIFBOD + 1
      DO 23 N=1,LBP1
      DO 23 I=NR,NT
   23 SIG(I,N) = 0.0
      LBP2 = LBP1 + 1
      DO 24 N = LBP2,NSIG
   24 READ(4) (SIG(I,N),I=NR,NT)
C    ***   SIGMAS HERE HAVE BECOME THE INPUT PV
      DO 26 N = LBP2,NSIG
      DO 26 I=NR,NT
   26 SIG(I,N) = SIG(I,N) /(-FOURPI)
      REWIND 4
C                  * NO. OF MIDPOINTS LOOP
   25 DO 100 I=1,NT
C                  * RFAD MATRICES A,B,Z
      READ (9) (A(J),J=1,NT),(B(J),J=1,NT),(7(J),J=1,NT)
C                  * NO. OF FLOWS LOOP
      N1=0
      DO 70 N=1,NSIG
      N1=N1+2
      SN=0.0
      ST=0.0
      SP=0.0
C                  * NO. OF ELEMENTS LOOP
      DO 30 J=1,NT
      SN=SN+A(J)*SIG(J,N)
      ST=ST+B(J)*SIG(J,N)
      IF(FLG23 .GT. 0)Z(J) = 0.0
   30 SP=SP+Z(J)*SIG(J,N)
      IF(FLG22.GT.0) GO TO 68
      IF (FLG12.EQ.0) GU TO 40
      XN(I,N)=SN
      PHT(I,N)=SP-RB(I,N1-1)
      GO TO 50
   40 XN(I,N)=SN-RB(I,N1-1)                                                 AXIS  106
      PHI(I,N)=SP                                                           AXIS  107
   50 IF (FLG11.EQ.0) GO TO 60                                              AXIS  108
      T(I,N)=ST                                                             AXIS  109
      GO TO 65                                                              AXIS  110
   60 T(I,N)=ST+RB(I,N1)                                                    AXIS  111
   65 SUMM(N)=SUMM(N)+PHI(I,N)*Y2(I)*RB(I,N1-1)*DELS(I)                     AXIS  112
      CP(I,N)=1.-T(I,N)**2                                                  AXIS  113
      GO TO 70                                                              AXIS  114
   68 XN(I,N) = SN                                                          AXIS  115
      PHI(I,N) = SP                                                         AXIS  116
      T(I,N) = ST                                                           AXIS  117
      CP(I,N) = 1.0 - T(I,N)**2                                             AXIS  118
   70 CONTINUE                                                              AXIS  119
      IF(FLG08.EQ.0) GO TO 100                                              AXIS  120
      WRITE (6,80) I,(A(J),J=1,NT)                                          AXIS  121
   80 FORMAT (1H0 13H MATRIX A ROW I6/ (1H 10F10.5))                        AXIS  122
      WRITE (6,85) I,(B(J),J=1,NT)                                          AXIS  123
   85 FORMAT (1H0 13H MATRIX B ROW I6/ (1H 10F10.5))                        AXIS  124
      WRITE (6,90) I,(Z(J),J=1,NT)                                          AXIS  125
   90 FORMAT (1H0 13H MATRIX Z ROW I6/ (1H 10F10.5))                        AXIS  126
  100 CONTINUE                                                              AXIS  127
      IF (MN.EQ.0.0) GO TO 130                                              AXIS  128
C                * MACH NO. ADJUSTMENT                                      AXIS  129
      D1=MN*MN                                                              AXIS  130
      D2=1.-D1                                                              AXIS  131
      D3=SQRT(D2)                                                           AXIS  132
      D4=.7*D1                                                              AXIS  133
      D5=.2*D1                                                              AXIS  134
      DO 120 N=1,NSIG                                                       AXIS  135
      DO 120 I=1,NT                                                         AXIS  136
      TX=(T(I,N)*COSA(I)-1.)/D2+1.                                          AXIS  137
      TY = ( T(I,N) * SINA(I) ) / D3                                        AXIS  138
      T(I,N)=SQRT(TX*TX+TY*TY)                                              AXIS  139
  120 CP(I,N)=((1.+D5*(1.-T(I,N)**2))**3.5-1.)/D4                           AXIS  140
```

```
C                * ELIMINATE MACH NO EFFECT FOR PRINTOUT
      DO 122 I=1,NT
  122 X1(I)=X1(I)*D3
      N=0
      J1=0
      DO 126 K=1,NB
      M=N+1
      N=N+ND(K)-1
      DO 124 J=M,N
      J1=J1+1
      T1=X1(J1+1)-X1(J1)
      T2=Y1(J1+1)-Y1(J1)
      X2(J)=(X1(J1+1)+X1(J1))/2.
      DELS(J)=SQRT(T1*T1+T2*T2)
      COSA(J)=T1/DFLS(J)
  124 SINA(J)=T2/DELS(J)
  126 J1=J1+1
C                * PRINT AXIS FLOW (ON-BODY) OUTPUT
  130 DO 250 L=1,NSIG
      KA = L
      IF (FLG16.LE.0) KA=L-1
      IF(FLG22 .GT. 0    .OR. FLG23 .GT. 0 )KA = L
      IF(FLG22.GT.0)GO TO 136
      SUMM(L)=-6.2831853*SUMM(L)
      DO 135 J = 1, NT
  135 SUMTDS(L) = SUMTDS(L) + T(J,L)*DFLS(J)
  136 I = 1
      J=1
      REWIND 1
      REWIND 3
      M=1
      N=ND(M)
      NSM=N-1
      XB(1)=X1(1)
      UB(1)=.9999999
      YB(1)=Y1(1)
      DO 138 KK=1,NSM
      XB(KK+1)=X2(KK)
      YB(KK+1)=Y2(KK)
  138 UB(KK+1)=CP(KK,L)
      WRITE(3) N
      WRITF(3) (XB(K),K=1,N)
      WRITF(3) (YB(K),K=1,N)
      WRITE(3) (UB(K),K=1,N)
      IF(ITER .GE. 1) GO TO 139
      WRITE(15) N
      WRITE(15) (X1(K),K=1,N)
      WRITE(15) (Y1(K),K=1,N)
  139 CONTINUE
      LCTR=22
  140 WRITE(6,150)HEDR,CASE,PSF
  150 FORMAT (1H1 25X, 26HDOUGLAS  AIRCRAFT  COMPANY /
     1        28X, 21HLONG  BEACH  DIVISION ///
     2  6X,10A6,4X,10HCASE NO.  A6,10H    PSF = ,A4  //)
      IF (FLG22.GT.0) GO TO 178
      IF (L.GT.1.OR.FLG16.NE.0) GO TO 170
      WRITE (6,160)
  160 FORMAT (1H 34H ON-BODY UNIFORM AXISYMMETRIC FLOW )
      GO TO 190
  170 IF (TYPEA(KA).GE.0.0) GO TO 175
      WRITE (6,172)
  172 FORMAT (1H 44H FLOW GENERATOR * ROTATING BODY * TYPE ERROR )
  175 IF (NUNA(KA).EQ.123456) WRITE (6,177)
  177 FORMAT (27H  ON-BODY STRIP VORTEX FLOW)
  178 IF (NUNA(KA).NE.123456) WRITE(6,180)NUNA(KA)
  180 FORMAT (1H 42H ON-BODY NON-UNIFORM AXISYMMETRIC FLOW NO. I8)
  190 WRITE (6,200)
  200 FORMAT (1H 5X 24H TRANSFORMED COORDINATES //
     1          12X 1HX 13X 1HY 13X            2HT1 12X 2HCP  9X 5HSIN A
     2          6X 5HCOS A 7X 5HSIGMA 11X 1HN 13X 3HPHI //)
  210 WRITF (6,220) I,X1(I),Y1(I),X2(J),Y2(J),          T(J,L),CP(J,L),
     1             STNA(J),COSA(J),SIG(J,L),XN(J,L),PHI(J,L)
  220 FORMAT (1H I3,2F14.7/ 4X 4F14.7,2F11.5,3F14.7)
      I=I+1
      J=J+1
      IF (I.EQ.N) GO TO 230
      IF (I.LF.LCTR) GO TO 210
      LCTR=LCTR+22
      GO TO 140
  230 M=M+1
      N=N+ND(M)
      WRITE (6,240) I,X1(I),Y1(I)
  240 FORMAT (1H I3, 2F14.7 //)
      I=I+1
      IF ( J - NT ) 210, 242, 242
  242 IF(FLG22.GT.0) GO TO 250
      WRITE(6,244) SUMM(L), SUMV, SUMTDS(L)
  244 FORMAT (1H0 10X 13H ADDED MASS =F12.7, 4X 9H VOLUME = F12.7,
     A         5X 18HSUM (T)(DELTA S) = F12.7 )
  250 CONTINUE
  252 LL = 1
      IF(FLG23 .GT. 0)CALL COMBO(LL)
C     IF(FLG05 .EQ. 0)RETURN
      IF(FLG05 .EQ. 0) GO TO 700
C                  * OFF-BODY POINT
  253 IF (FLG15.LE.0) GO TO 258
      M = 0
      DO 254 I = 1, NB
  254 IF (NLF(I) .LE. 0) M = M + 1
      IF ( M .EQ. 0 ) GO TO 258
      MM = NNU + 1
      IF(FLG23 .GT. 0)MM = MM + NNU
      DO 255 I = 1, MM
  255 READ (4)
      IF (FLG22.GT.0)READ(4)
      DO 256 J = 1, M
  256 READ(4) (RB(I,J),I = 1,NP), (T3(I,J),I = 1,NP)
      REWIND 4
  258 DO 300 I = 1, NP
      L=0
C               * READ MATRICES X,Y,Z
      READ (9) (A(J),J=1,NT),(B(J),J=1,NT),(7(J),J=1,NT)
C               * NO. OF FLOW
      DO 300 N=1,NSIG
      KA=N
      IF (FLG16.LE.0) KA=N-1
      SX=0.0
      SY=0.0
      SP=0.0
C               * NO. OF ELEMENTS LOOP
      DO 260 J=1,NT
      SX=SX+A(J)*SIG(J,N)
      SY=SY+B(J)*SIG(J,N)
      IF(FLG23 .GT. 0)Z(J) = 0.0
  260 SP=SP+Z(J)*SIG(J,N)
      PHI(I,N)=SP
      IF (FLG22.GT.0) GO TO 270
      IF (FLG11.GT.0) GO TO 270
      IF (N.NE.1.OR.FLG16.GT.0) GO TO 262
      VX(I,N) = SX+1.
      GO TO 280
  262 IF (NUNA(KA).NE.123456) GO TO 270
      L=L+1
      VX(I,N)=SX+RB(I,L)
      VY(I,N)=SY+T3(I,L)
      GO TO 300
  270 VX(I,N) = SX
  280 VY(I,N) = SY
  300 CONTINUE
      IF (MN.EQ.0.0) GO TO 330
```

```
C                * MACH NO. ADJUSTMENT                                        AXIS  281
      DO 320 N=1,NSIG                                                         AXIS  282
      DO 320 I=1,NP                                                           AXIS  283
      VY(I,N)=VY(I,N)/D3                                                      AXIS  284
  320 VX(I,N)=(VX(I,N)-1.)/D2+1.                                              AXIS  285
      DO 322 I = 1, NP                                                        AXIS  286
  322 XP(I)=XP(I)*D3                                                          AXIS  287
C                * COMPUTE VT AND THETA                                       AXIS  288
  330 DO 335 N=1,NSIG                                                         AXIS  289
      DO 335 I=1,NP                                                           AXIS  290
      VT(I,N)=SQRT(VX(I,N)**2+VY(I,N)**2)                                     AXIS  291
  335 TH(I,N)=ATAN2(VY(I,N),VX(I,N)) * 57.29578                               AXIS  292
C                * PRINT AXIS FLOW (OFF-BODY) OUTPUT                          AXIS  293
      DO 450 L=1,NSIG                                                         AXIS  294
      KA = L                                                                  AXIS  295
      IF(FLG16 .LE. 0)KA = L - 1                                              AXIS  296
      IF(FLG22 .GT. 0   .OR.    FLG23 .GT. 0) KA = L                          AXIS  297
      I=1                                                                     AXIS  298
      LCTR=45                                                                 AXIS  299
  340 WRITE(6,150)HEDR,CASE,PSF                                               AXIS  300
      IF (L.GT.1.OR.FLG16.NE.0) GO TO 370                                     AXIS  301
      IF(FLG22.GT.0) GO TO 378                                                AXIS  302
      WRITE (6,360)                                                           AXIS  303
  360 FORMAT (1H 35H OFF-BODY UNIFORM AXISYMMETRIC FLOW )                     AXIS  304
      GO TO 390                                                               AXIS  305
  370 IF (TYPEA(KA).GE.0.) GO TO 375                                          AXIS  306
      WRITE (6,172)                                                           AXIS  307
  375 IF (NUNA(KA).EQ.123456) WRITE (6,377)                                   AXIS  308
  377 FORMAT (28H  OFF-BODY STRIP VORTEX FLOW)                                AXIS  309
  378 IF (NUNA(KA).NE.123456) WRITE (6,380) NUNA(KA)                          AXIS  310
  380 FORMAT (1H 43H OFF-BODY NON-UNIFORM AXISYMMETRIC FLOW NO. I8)           AXIS  311
  390 WRITE (6,400)                                                           AXIS  312
  400 FORMAT (1H 5X, 24H TRANSFORMED COORDINATES //                           AXIS  313
     1          12X 1HX 13X 1HY 13X         2HVX 12X 2HVY 12X 2HVT 10X        AXIS  314
     2          5HTHETA 11X 3HPHI //)                                         AXIS  315
  410 WRITE (6,420) I,XP(I),YP(I),                VX(I,L),VY(I,L),VT(I,L),          AXIS  316
     1 TH(I,L), PHI(I,L)                                                            AXIS  317
  420 FORMAT (1H I3, 7F14.7)                                                        AXIS  318
      I=I+1                                                                         AXIS  319
      IF (I.GT.NP) GO TO 450                                                        AXIS  320
      IF (I.LE.LCTR) GO TO 410                                                      AXIS  321
      LCTR=LCTR+45                                                                  AXIS  322
      GO TO 340                                                                     AXIS  323
  450 CONTINUE                                                                      AXIS  324
  500 LL = 0                                                                        AXIS  325
      IF(FLG23 .GT. 0)CALL COMBU(LL)                                                AXIS  326
C     RETURN                                                                        AXIS  327I
  700 CONTINUE                                                                      AXIS  328C
      END                                                                           AXIS  329
      SUBROUTINE COMBO(LL)                                                      COMB 001
C                                                                               COMB 002
      COMMON / IPSF /   PSF                                                     COMB 003
      COMMON /COMBIN/CHAY(2)                                                    COMB 004
      COMMON       HEDR(10)  ,CASE       ,NB         ,NNU                       COMB 005
     1            ,FLG03     ,FLG04      ,FLG05      ,FLG06      ,FLG07         COMB 006
     2            ,FLG08     ,FLG09      ,FLG10      ,FLG11      ,FLG12         COMB 007
     3            ,FLG13     ,FLG14      ,FLG15      ,FLG16      ,FLG17         COMB 008
     4            ,FLG18     ,FLG19      ,FLG20      ,FLG21      ,FLG22         COMB 009
     5            ,FLG23     ,FLG24      ,FLG25      ,FLG26      ,FLG27         COMB 010
      COMMON   NT,        ND(11),    MN,        NUNA(5),  TYPEA(5),             COMB 011
     1         NER1,      NER2,      NMA,       NSIGA,    NSIGC,                COMB 012
     2         NUNC(5),   TYPFC(5),  NLF(11),   IEC,      NSIGEC,               COMB 013
     3         TYPFEC(5),NUNFC(5)                                               COMB 014
C     DOUBLE PRECISION  HEDR, CASE                                              COMB 015I
      INTEGER      FLG03     ,FLG04      ,FLG05      ,FLG06      ,FLG07         COMB 016
     1            ,FLG08     ,FLG09      ,FLG10      ,FLG11      ,FLG12         COMB 017
     2            ,FLG13     ,FLG14      ,FLG15      ,FLG16      ,FLG17         COMB 018
     3            ,FLG18     ,FLG19      ,FLG20      ,FLG21      ,FLG22         COMB 019
     4            ,FLG23     ,FLG24      ,FLG25      ,FLG26      ,FLG27         COMB 020
      REAL         MN                                                           COMB 021
C                                                                               COMB 022
      COMMON /C4/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),          COMB 023
     1              SINA(100),COSA(100),XP(100),  YP(100)                       COMB 024
      COMMON /TC/   RB(100,10),    SIG(100,5),    A(100),   B(100),             COMB 025
     1              Z(100),        PHI(100,5),    XN(100,5),T(100,5),           COMB 026
     2              T3(100,5),     NSIG,          NP,       NI,                 COMB 027
     3              SUMV,          SUMM(5)                                      COMB 028
      DIMENSION C(2,2), DV(2),           TFIRST(2,5),TLAST(2,5),TSUM(2,5)       COMB 029
      DIMENSION  CP(100)                                                        COMB 030
      EQUIVALENCE ( CP(1),A(1) )                                                COMB 031
      EQUIVALENCE (  DV(1),CHAY(1)   )                                          COMB 032
      DIMENSION VX(100,5),VY(100,5),VT(100,5)                                   COMB 033
      EQUIVALENCE  (VX(1,1),XN(1,1)),(VY(1,1),T(1,1)),                          COMB 034
     1   (VT(1,1),T3(1,1))                                                      COMB 035
```

```
C
C   **    ICNT WILL BE THE NUMBER OF LIFTING BODIES
C   *    NFLOW WILL BE THE NUMBER OF FLOWS
C
        ICNT = 0
        DO 10 K=1,NB
     10 IF( NLF(K) .LE. 0)ICNT=ICNT+1
        NFLOW = 1 + 2*ICNT
        IF(LL .EQ. 0)GO TO 1000
        READ(5,4)( DV(I),I=1,ICNT )
      4 FORMAT (6F10.0)
        WRITE(6,6) (DV(I),I=1,ICNT)
      6 FORMAT(1H1,42H THE INPUT DV FOR COMBINATION SOLUTION ARE /
       1   (2X,6F10.4)   )
C
C    ***    IPTCNT WILL BE THE LAST MIDPOINT ON BODY K
C
        IPTCNT = 0
C   *    ILIFT WILL BE THE LIFTING BODY NUMBER
        ILIFT = 0
        DO 100 K=1,NB
        IPTCNT = IPTCNT + ND(K) - 1
        IFIRST = IPTCNT - ND(K) + 2
        IF ( NLF(K) .GT. 0 )GO TO 100
        ILIFT = ILIFT + 1
C
        DO 50 J=1,NFLOW
        TFIRST(ILIFT,J) = T(IFIRST,J)
        TLAST(ILIFT,J) = T(IPTCNT,J)
     50 TSUM(ILIFT,J)   = TFIRST(ILIFT,J) + TLAST(ILIFT,J)
    100 CONTINUE
C
C   **    IPVBOD WILL BE 1ST PRESCRIBED VORTICITY FLOW
C   *    LASTSV WILL BE LAST STRIP VORTEX FLOW
C
```

```
      IPVBOD = 2+ICNT
      LASTSV = IPVROD - 1
      DO 300 I=1,ICNT
      JCNT = 0
      DV(I)= DV(I) - TSUM(I,1)
C
      DO 200 J=IPVBOD,NFLOW
  200 DV(I) = DV(I) - TSUM(I,J)
C
      DO 250 J=2,LASTSV
      JCNT = JCNT + 1
  250 C(I,JCNT) = TSUM(I,J)
  300 CONTINUE
C
C
      CALL SOLCOM( DV, C, ICNT )
      DO 500 I = 1,NT
C    ***   ADD PV FI OWS TO AXIS FLOW FOR COMBTNATION SOLUTION   ON BODY
      JCNT = 0
      DO 350 J=IPVBOD,NFLOW
      XN(I,1) = XN(I,1) + XN(I,J)
  350 T(I,1) = T(I,1) + T(I,J)
C    ***   ADD K * STRIP VORTEX BODY VELOCITY FOR COMBINATION SOLUTION
      DO 400 J = 2,LASTSV
      JCNT = JCNT + 1
      XN(I,1) = XN(I,1) + CHAY(JCNT)* XN(I,J)
  400 T(I,1) = T(I,1) + CHAY(JCNT)*T(I,J)
      CP(I) = 1.0  - T(I,1)**2
  500 CONTINUE
      I=1
      J=1
      M=1
      N=ND(M)
      LCTR = 22
  540 WRITE(6,550)HEDR,CASE,PSF
```

```
550 FORMAT(1H1,25X, 26HDOUGLAS  AIRCRAFT  COMPANY /
   1       28X, 21HLONG  BEACH  DIVISION ///
   2 6X,10A6, 4X,10HCASE NO.  A6, 9H   PSF = ,A4  // )
    WRITE(6,560)
560 FORMAT( 1H ,21H COMBINATION SOLUTION  )
    WRITE(6,700)
700 FORMAT(1H 5X 24H TRANSFORMED COORDINATES //
   1 12X,1HX 13X 1HY 13X 2HT1 12X 2HCP 9X 5HSIN A 6X 5HCOS A 11X 1HN
   2 // )
710 WRITE(6,720)I,X1(I),Y1(I),X2(J),Y2(J),T(J,1),CP(J),SINA(J),COSA(J)
   2         ,XN(J,1)
720 FORMAT(1H I3,2F14.7 / 4X 4F14.7,2F11.5,F14.7 )
    I=I+1
    J=J+1
    IF( I  .EQ. N) GO TO 730
    IF( I  .LE. LCTR ) GO TO 710
    LCTR = LCTR + 22
    GO TO 540
730 M=M+1
    N=N+ND(M)
    WRITE(6,740)I,X1(I),Y1(I)
740 FORMAT( 1H ,I3, 2F14.7 // )
    I=I+1
    IF( J .LT. NT)GO TO 710
    M1 = 0
    N1 = 0
    DO 800 K = 1,NB
    M1 = N1 + 1
    N1 = N1 + ND(K) - 1
    CIRC = 0.0
    THRUST = 0.0
    DO 790 I = M1,N1
    CIRC = CIRC + ( T(I,1) * DELS(I) )
790 THRUST = THRUST + ( Y2(I) * CP(I) * SINA(I) * DELS(I) )
    THRUST = -6.283186 * THRUST
```

```
      WRITF(6,795)K,CIRC,THRUST
  795 FORMAT(///13H    BODY NO. ,I4,5X,14HCIRCULATION = ,F14.7,5X,
     1      9HTHRUST = ,F14.7)
  800 CONTINUE
      RETURN
C   ***   OFF BODY COMBINATION SOLUTION
 1000 IPVBOD = 2 + ICNT
      LASTSV = IPVBOD - 1
      DO 1500 I=1,NP
      JCNT = 0
C   ***   ADD PV FLOWS TO AXIS FLOW FOR COMBINATION SOLUTION   OFF BODY
      DO 1350 J = IPVBOD,NFLOW
      VX(I,1) = VX(I,1) + VX(I,J)
 1350 VY(I,1) = VY(I,1) + VY(I,J)
C   ***   ADD K * STRIP VORTEX OFF BODY VELOCITY
      DO 1400 J=2,LASTSV
      JCNT = JCNT + 1
      VX(I,1) = VX(I,1) + CHAY(JCNT) * VX(I,J)
 1400 VY(I,1) = VY(I,1) + CHAY(JCNT) * VY(I,J)
      VT(I,1) = SQRT( VX(I,1)**2 + VY(I,1)**2 )
 1500 CONTINUE
      I = 1
      LCTR = 45
 1540 WRITE(6,550)HEDR,CASE,PSF
      WRITF(6,1560)
 1560 FORMAT(1H ,31H COMBINATION SOLUTION  OFF BODY  )
      WRITF(6,1700)
 1700 FORMAT(1H ,5X,24H TRANSFORMED COORDINATES //
     1   12X,1HX,13X,1HY,13X,2HVX,12X,2HVY,12X,2HVT //)
 1710 WRITF(6,1720)I,XP(I),YP(I),VX(I,1),VY(I,1),VT(I,1)
 1720 FORMAT(1H ,I3,5F14.7)
      I = I + 1
      IF(I .GT. NP)GO TO 1750
      IF(I .LE. LCTR )GO TO 1710
      LCTR = LCTR + 45
      GO TO 1540                                                        COMB  176
 1750 CONTINUE                                                          COMB  177
      RETURN                                                            COMB  178
      END                                                               COMB  179
      SUBROUTINE SOLCOM( DV,  A   , ICNT)
C
C  **   NOTE THAT THIS SUBROUTINE SOLVES ONLY A 1X1 OR 2X2 MATRIX
C  **   IT IS SEPARATED SO THAT IF PROGRAM IS EVER INLARGED, THIS
C  **   IS WHERE THE MATRIX SOLUTION FOR THE COMBINATION PART OF
C  **   THE PROGRAM WILL GO.  THE MATRICES HAVE BEEN FORMED GENERALLY
C  **   IN SUBROUTINE COMBO.
C
      DIMENSION DV(2), A(2,2)
      IF(ICNT .EQ. 2) GO TO 20
      DV(1) = DV(1) / A(1,1)
      RETURN
   20 DV(1) = (DV(1) - (A(1,2)/A(2,2) ) * DV(2)  ) /
     1        (  A(1,1)  -  A(1,2)*A(2,1) / A(2,2)  )
      DV(2) =( DV(2) - A(2,1)*DV(1) ) / A(2,2)
      RETURN
      END
      OVERLAY(AXSY,4,2)                                                 CROS 001C
      PROGRAM CROSS                                                     CROS 002C
C     SUBROUTINE CROSS                                                  CROS 003I
C                                                                       CROS 004
C              * COMPUTE CROSS FLOW VELOCITY COMPONENTS AND PRINT       CROS 005
C                                                                       CROS 006
      COMMON /IPSF/ PSF                                                 CROS 007
      COMMON       HEDR(10)   ,CASE       ,NB         ,NNU              CROS 008
     1            ,FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07 CROS 009
     2            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12 CROS 010
     3            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17 CROS 011
     4            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22 CROS 012
     5            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27 CROS 013
      COMMON    NT,        ND(11),    MN,        NUNA(5),   TYPEA(5),   CROS 014
     1          NER1,      NER2,      NMA,       NSIGA,     NSIGC,      CROS 015
     2          NUNC(5),   TYPEC(5),  NLF(11),   IEC,       NSIGEC,     CROS 016
     3          TYPEEC(5),NUNEC(5)                                      CROS 017
C     DOUBLE PRECISION  HEDR, CASE                                      CROS 018I
      INTEGER      FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07 CROS 019
     1            ,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12 CROS 020
     2            ,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17 CROS 021
     3            ,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22 CROS 022
     4            ,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27 CROS 023
      REAL         MN                                                   CROS 024
C                                                                       CROS 025
      COMMON /C4/    X1(100),   Y1(100),   X2(100),   Y2(100),   DELS(100), CROS 026
     1               SINA(100),COSA(100),XP(100),   YP(100)             CROS 027
      COMMON /TC/    RB(100,10),     SIG(100,5),     A(100),    B(100), CROS 028
     1               Z(100),         PHI(100,5),     XN(100,5),T(100,5), CROS 029
     2               T3(100,5),      NSIG,           NP,        NI,     CROS 030
     3               SUMV,           SUMM(5)                            CROS 031
C                                                                       CROS 032
      DIMENSION VX(100,5),VY(100,5),VZ(100,5),T2(100,5)                 CROS 033
C                                                                       CROS 034
      EQUIVALENCE ( VX(1,1), XN(1,1) ), ( VY(1,1), T(1,1) ),           CROS 035
     1   (VZ(1,1), T3(1,1) ),  (T2(1,1), T(1,1) )
C
C                 * START
      IF (FLG08.EQ.0) GO TO 10
C                 * TITLE FOR MATRIX PRINT
      WRITF(6,150)HEDR,CASE,PSF
      WRITF (6,8)
    8 FORMAT (1H 36H MATRICES A,B,Z BY ROWS * CROSS FLOW //)
C                 * RFAD CROSS SIGMAS
   10 DO 20 N=1,NSIG
      SUMM(N)=0.0
   20 READ (3) (SIG(I,N),I=1,NT)
C                 * NO. OF MIDPOINTS LOOP
      DO 100 I=1,NT
C                 * READ MATRICES A,B,Z
      READ (10) (A(J),J=1,NT),(B(J),J=1,NT),(Z(J),J=1,NT)
C                 * NO. OF FLOWS LOOP
      M=0
      DO 70 N=1,NSIG
      M=M+2
      SA=0.0
      SB=0.0
      SZ=0.0
C                 * NO. OF ELFMENTS LOOP
      DO 30 J=1,NT
      SA=SA+A(J)*SIG(J,N)
      SB=SB+B(J)*SIG(J,N)
   30 SZ=SZ+Z(J)*SIG(J,N)
C                 * INITIALIZF UNIFORM OR NON-UNIFORM PARAMETERS
      IF (FLG21.GT.0) GO TO 38
      IF (N.EQ.1.AND.FLG17.LE.0) GO TU 35
      C1=RB(I,M)
      C2 = -RB(I,M-1)
      C3=0.0
      GO TO 40
   35 C1=SINA(I)
      C2=COSA(I)
      C3=1.
      GO TO 40
   38 C1 = 0.0
      C2 = 0.0
      C3 = 0.0
   40 IF (FLG12.EQ.0) GO TO 45
C                * OPTION FOR Z (PHI) MATRIX SOLUTION
      XN(I,N) = SA
      PHI(I,N) = Y2(I) * SZ
      GO TO 50
C                * REGULAR A MATRIX SOLUTION
   45 PHI(I,N)=Y2(I)*SZ
      XN(I,N)=SA+C2
   50 IF (FLG11.EQ.0) GO TO 55
C                * OPTION PERTURBATIONS
      T2(I,N)=SB
      T3(I,N)=SZ
      GO TO 60
   55 T2(I,N)=SB+C1
      T3(I,N)=SZ+C3
   60 IF(FLG21.GT.0) GO TO 70
      SUMM(N) = SUMM(N) + PHI(I,N) * Y2(I) * C2 * DELS(I)
   70 CONTINUE
      IF (FLG08.EQ.0) GO TO 100
      WRITE (6,80) I,(A(J),J=1,NT)
   80 FORMAT (1H0 13H MATRIX A ROW I6/  (1H 10F10.5))
      WRITE (6,85) I,(B(J),J=1,NT)
   85 FORMAT (1H0 13H MATRIX B ROW I6/ (1H 10F10.5))
      WRITE (6,90) I,(Z(J),J=1,NT)
   90 FORMAT (1H0 13H MATRIX Z ROW I6/ (1H 10F10.5))
  100 CONTINUE
C                * PRINT CROSS FLOW (ON-BODY) OUTPUT
  130 DO 250 L=1,NSIG
```

```
      KC = L
      IF (FLG17.LE.0) KC=L-1
      IF(FLG21.GT.0) GO TO 138
      SUMM(L) = 3.141593  *SUMM(L)
  138 I = 1
      J=1
      M=1
      N=ND(M)
      LCTR=22
  140 WRITE(6,150)HEDR,CASE,PSF
  150 FORMAT (1H1 25X, 26HDOUGLAS  AIRCRAFT  COMPANY /
     1       28X, 21HLONG  BEACH  DIVISION ///
     2  6X,10A6,4X,10HCASE NO.  A6,10H    PSF = ,A4  //)
      IF (FLG22.GT.0) GO TO 175
      IF (L.GT.1.OR.FLG17.NE.0) GO TO 170
      WRITE (6,160)
  160 FORMAT (1H 27H ON-BODY UNIFORM CROSS FLOW )
      GO TO 190
  170 IF (TYPEC(KC).GE.0.) GO TO 175
      WRITE (6,172)
  172 FORMAT (1H 31H FLOW GENERATOR * ROTATING BODY )
  175 WRITE (6,180) NUNC(KC)
  180 FORMAT (1H 35H ON-BODY NON-UNIFORM CROSS FLOW NO. I8)
  190 WRITE (6,200)
  200 FORMAT (1H 5X 24H TRANSFORMED COORDINATES //
     1        12X 1HX 13X 1HY 13X           2HT2 12X 2HT3   9X 5HSIN A
     2        6X 5HCOS A 7X 5HSIGMA 11X 1HN 13X 3HPHI //)
  210 WRITE (6,220) I,X1(I),Y1(I),X2(J),Y2(J),          T2(J,L),T3(J,L),
     1            SINA(J),COSA(J),SIG(J,L),XN(J,L),PHI(J,L)
  220 FORMAT (1H I3,2F14.7/ 4X 4F14.7,2F11.5,3F14.7)
      I=I+1
      J=J+1
      IF (I.EQ.N) GO TO 230
      IF (I.LE.LCTR) GO TO 210
      LCTR=LCTR+22
      GO TO 140
  230 M=M+1
      N=N+ND(M)
      WRITE (6,240) I,X1(I),Y1(I)
  240 FORMAT (1H I3, 2F14.7 //)
      I=I+1
      IF(J.GT.NT)GO TO 242
      GO TO 210
  242 IF(FLG22.GT.0)GO TO 250
      WRITE(6,244) SUMM(L), SUMV
  244 FORMAT (1H0 10X,14H ADDED MASS = F12.7, 4X,10H VOLUME = F12.7)
  250 CONTINUE
  252 IF (FLG05.EQ.0) RETURN
C              * OFF-BODY POINT
      DO 300 I=1,NP
C              * READ MATRICES X,Y,Z
      READ (10) (A(J),J=1,NT),(B(J),J=1,NT),(Z(J),J=1,NT)
C              * NO. OF FLOW
      DO 300 N=1,NSIG
      SX=0.0
      SY=0.0
      SP=0.0
C              * NO. OF ELEMENTS LOOP
      DO 260 J=1,NT
      SX=SX+A(J)*SIG(J,N)
      SY=SY+B(J)*SIG(J,N)
  260 SP=SP+Z(J)*SIG(J,N)
      VX(I,N)=SX
      PHI(I,N)=YP(I)*SP
      IF (FLG22.GT.0) GO TO 270
      IF (FLG11.GT.0.OR.N.NE.1.OR.FLG17.GT.0) GO TO 270
      VY(I,N)=SY+1.
      VZ(I,N)=SP+1.
      GO TO 300
C              * PERTURBATION OR NON-UNIFORM VY,VZ
```

```
  270 VY(I,N)=SY
      VZ(I,N)=SP
  300 CONTINUE
C               * PRINT CROSS FLOW (OFF-BODY) OUTPUT
  330 DO 450 L=1,NSIG
      KC = L
      IF (FLG17.LE.0) KC=L-1
      I=1
      LCTR=45
  340 WRITE(6,150)HEDR,CASF,PSF
      IF (FLG22.GT.0) GO TO 375
      IF (L.GT.1.OR.FLG17.NE.0) GO TO 370
      WRITE (6,360)
  360 FORMAT (1H 28H OFF-BODY UNIFORM CROSS FLOW )
      GO TO 390
  370 IF (TYPEC(KC).GE.0.) GO TO 375
      WRITE (6,172)
  375 WRITE (6,380) NUNC(KC)
  380 FORMAT (1H 36H OFF-BODY NON-UNIFORM CROSS FLOW NO. I8)
  390 WRITE (6,400)
  400 FORMAT (1H 5X, 24H TRANSFORMED COORDINATES //
     1 12X 1HX 13X 1HY 13X 2HVX 12X 2HVY 12X 2HVZ 12X 3HPHI //)
  410 WRITE (6,420) I,XP(I),YP(I),VX(I,L),VY(I,L),VZ(I,L),PHI(I,L)
  420 FORMAT (1H I3, 6F14.7)
      I=I+1
      IF (I.GT.NP) GO TO 450
      IF (I.LE.LCTR) GO TO 410
      LCTR=LCTR+45
      GO TO 340
  450 CONTINUE
  500 CONTINUE
C     RETURN
      END
      OVERLAY(AXSY,4,3)                                                 EXCR  001C
      PROGRAM EXCROS                                                    EXCR  002C
C     SUBROUTINE EXCROS                                                 EXCR  003I
C***   ***COMPUTE EXTRA CROSS FLOW VELOCITY COMPONENTS AND PRINT        EXCR  004
      COMMON /IPSF/ PSF                                                 EXCR  005
      COMMON       HEDR(10)  ,CASE      ,NB        ,NNU                 EXCR  006
     1             ,FLG03     ,FLG04     ,FLG05     ,FLG06     ,FLG07   EXCR  007
     2             ,FLG08     ,FLG09     ,FLG10     ,FLG11     ,FLG12   EXCR  008
     3             ,FLG13     ,FLG14     ,FLG15     ,FLG16     ,FLG17   EXCR  009
     4             ,FLG18     ,FLG19     ,FLG20     ,FLG21     ,FLG22   EXCR  010
     5             ,FLG23     ,FLG24     ,FLG25     ,FLG26     ,FLG27   EXCR  011
      COMMON   NT,        ND(11),   MN,        NUNA(5),  TYPEA(5),      EXCR  012
     1         NER1,      NER2,     NMA,       NSIGA,    NSIGC,         EXCR  013
     2         NUNC(5),   TYPEC(5), NLF(11),   IFC,      NSIGEC,        EXCR  014
     3         TYPEEC(5),NUNEC(5)                                       EXCR  015
C     DOUBLE PRECISION  HEDR, CASE                                      EXCR  016I
      INTEGER      FLG03     ,FLG04     ,FLG05     ,FLG06     ,FLG07    EXCR  017
     1             ,FLG08     ,FLG09     ,FLG10     ,FLG11     ,FLG12   EXCR  018
     2             ,FLG13     ,FLG14     ,FLG15     ,FLG16     ,FLG17   EXCR  019
     3             ,FLG18     ,FLG19     ,FLG20     ,FLG21     ,FLG22   EXCR  020
     4             ,FLG23     ,FLG24     ,FLG25     ,FLG26     ,FLG27   EXCR  021
      REAL         MN                                                   EXCR  022
C                                                                       EXCR  023
      COMMON /C4/   X1(100),  Y1(100),  X2(100),  Y2(100),  DELS(100),  EXCR  024
     1              SINA(100),COSA(100),XP(100),  YP(100)               EXCR  025
      COMMON /TC/   RB(100,10),     SIG(100,5),     A(100),   B(100),   EXCR  026
     1              Z(100),         PHI(100,5),     XN(100,5),T(100,5), EXCR  027
     2              T3(100,5),      NSIG,           NP,       NI,       EXCR  028
     3              SUMV,           SUMM(5)                             EXCR  029
C                                                                       EXCR  030
      DIMENSION VX(100,5),VY(100,5),VZ(100,5),T2(100,5)                 EXCR  031
C                                                                       EXCR  032
      EQUIVALENCE ( VX(1,1), XN(1,1) ), ( VY(1,1), T(1,1) ),            EXCR  033
     1   (VZ(1,1), T3(1,1) ),  (T2(1,1), T(1,1) )                       EXCR  034
C                                                                       EXCR  035
```

```
      REWIND 8
      IF (FLG08.EQ.0) GO TO 10
C***    ***TITLE FOR MATRIX PRINT
      WRITE(6,150)HEDR,CASF,PSF
      WRITE (6,8)
    8 FORMAT (1H 42H MATRICES A,B,Z BY ROWS * EXTRA CROSS FLOW //)
C***    ***READ EXTRA CROSS SIGMAS
   10 DO 20 N = 1,NSIG
   20 READ (3) ( SIG(I,N),I = 1,NT )
C***    ***NO. OF MIDPOINTS LOOP
      DO 100 I = 1,NT
C***    ***READ MATRICES A,B,Z
C***    ***YOU MUST SOLVE POTENTIAL MATRIX FOR EXCROS
      READ (8) ( A(J),J = 1,NT),( B (J),J = 1,NT ), ( Z(J),J = 1,NT )
C***    ***NO. OF FLOWS LOOP
      M = 0
      DO 70 N = 1,NSIG
      M = M + 2
      SA = 0.0
      SB = 0.0
      SZ = 0.0
C***    ***NO. OF ELEMENTS LOOP
      DO 30 J = 1,NT
      SA = SA + A(J) * SIG(J,N)
      SB = SB + B(J) * SIG(J,N)
   30 SZ = SZ + Z(J) * SIG(J,N)
   40 T2(I,N) = SB
      T3(I,N) = SZ
      XN(I,N) = SA
      PHI(I,N) = Y2(I) * SZ / 2.0
   70 CONTINUE
      IF (FLG08.EQ.0) GO TO 100
      WRITE (6,80) I, (A(J),J = 1,NT)
   80 FORMAT (1H0 13H MATRIX A ROW I6/  (1H 10F10.5) )
      WRITE (6,85) I, (B(J),J = 1,NT)
   85 FORMAT (1H0 13H MATRIX B ROW I6/  (1H 10F10.5) )
      WRITE (6,90) I, ( Z(J),J = 1,NT)
   90 FORMAT (1H0 13H MATRIX Z ROW I6/  (1H 10F10.5) )
  100 CONTINUE
C***   ***PRINT EXTRA CROSS FLOW (ON BODY) OUTPUT
  130 DO 250 L = 1,NSIG
      KEC = L
      I = 1
      J = 1
      M = 1
      N = ND(M)
C***   ***M IS THE BODY NUMBER
C***   ***N IS THE NUMBER OF POINTS ON BODY M
      LCTR = 22
  140 WRITE(6,150)HEDR,CASE,PSF
  150 FORMAT (1H1 25X, 26HDOUGLAS  AIRCRAFT  COMPANY /
     1        28X, 21HLONG  BEACH  DIVISION ///
     2  6X,10A6,4X,10HCASE NO.  A6,10H    PSF = ,A4  //)
      IF (FLG22.GT.0) GO TO 160
      WRITE (6,155)NUNEC(KEC)
  155 FORMAT(41H ON-BODY NON-UNIFORM EXTRA CROSS FLOW NO. I8)
      GO TO 190
  160 WRITE  (6,162)
  162 FORMAT(68H  ON BODY   GENERATED (RESEP) BOUNDARY CONDITIONS   EXTR
     1A CROSS FLOW)
  190 WRITE (6,200)
  200 FORMAT (1H 5X 24H TRANSFORMED COORDINATES //
     1          12X 1HX 13X 1HY 13X              2HT2 12X 2HT3  9X 5HSIN A
     2          6X 5HCOS A 7X 5HSIGMA 11X 1HN 13X 3HPHI //)
  210 WRITE (6,220) I,X1(I),Y1(I),X2(J),Y2(J),        T2(J,L),T3(J,L),
     1              SINA(J),COSA(J),SIG(J,L),XN(J,L),PHI(J,L)
  220 FORMAT (1H I3,2F14.7/ 4X 4F14.7,2F11.5,3F14.7)
      I = I + 1
      J = J + 1
      IF (I.EQ.N) GO TO 230
      IF (I.LE.LCTR) GO TO 210                                                    EXCR  106
      LCTR = LCTR + 22                                                            EXCR  107
      GO TO 140                                                                   EXCR  108
  230 M = M + 1                                                                   EXCR  109
      N = N + ND(M)                                                               EXCR  110
      WRITE (6,240)I , X1(I), Y1(I)                                               EXCR  111
  240 FORMAT (1H I3,2F14.7 //)                                                    EXCR  112
      I = I + 1                                                                   EXCR  113
      IF (J.GE.NT) GO TO 250                                                      EXCR  114
      GO TO 210                                                                   EXCR  115
  250 CONTINUE                                                                    EXCR  116
C 252 IF (FLG05.EQ.0) RETURN                                                     EXCR  117I
  252 IF (FLG05.EQ.0) CONTINUE                                                   EXCR  118C
C***   ***OFF BODY POINTS                                                         EXCR  119
      DO 300 I = 1,NP                                                             EXCR  120
C***   ***READ MATRICES X,Y,Z                                                     EXCR  121
      READ (8) ( A(J),J=1,NT ),(B(J),J = 1,NT), ( Z(J),J = 1,NT )                 EXCR  122
      DO 300 N = 1,NSIG                                                           EXCR  123
      SX = 0.0                                                                    EXCR  124
      SY = 0.0                                                                    EXCR  125
      SP = 0.0                                                                    EXCR  126
C***   ***NUMBER OF ELEMENTS LOOP                                                 EXCR  127
      DO 260 J = 1,NT                                                             EXCR  128
      SX = SX + A(J) * SIG (J,N)                                                  EXCR  129
      SY = SY + B(J) * SIG (J,N)                                                  EXCR  130
  260 SP = SP + Z(J) * SIG(J,N)                                                   EXCR  131
      VX(I,N) = SX                                                                EXCR  132
      VY(I,N) = SY                                                                EXCR  133
      VZ(I,N) = SP                                                                EXCR  134
      PHI(I,N) = YP(I) * SP / 2.0                                                 EXCR  135
  300 CONTINUE                                                                    EXCR  136
C***   ***PRINT EXTRA CROSS FLOW (OFF-BODY) OUTPUT                                EXCR  137
  330 DO 450 L = 1,NSIG                                                           EXCR  138
      KEC = L                                                                     EXCR  139
      I = 1                                                                       EXCR  140
      LCTR = 45
  340 WRITE(6,150)HEDR,CASE,PSF
      IF (FLG22.GT.0) GO TO 355
      WRITF(6,350) NUNEC(KEC)
  350 FORMAT (43H  OFF BODY NON-UNIFORM EXTRA CROSS FLOW NO. I8)
      GO TO 390
  355 WRITE(6,357)
  357 FORMAT(68H OFF BODY   GENERATED (RFSEP) BOUNDARY CONDITIONS   FXTR
     1A CROSS FLOW)
  390 WRITE (6,400)
  400 FORMAT (1H 5X, 24H TRANSFORMFD COORDINATFS //
     1 12X 1HX 13X 1HY 13X 2HVX 12X 2HVY 12X 2HVZ 12X 3HPHI //)
  410 WRITE (6,420) I,XP(I),YP(I),VX(I,L),VY(I,L),VZ(I,L),PHI(I,L)
  420 FORMAT (1H I3, 6F14.7)
      I = I + 1
      IF (I.GT.NP)GO TO 450
      IF (I.LE.LCTR) GO TO 410
      LCTR = LCTR + 45
      GO TO 340
  450 CONTINUE
C     RETURN
  501 CONTINUE
      END
      OVERLAY(AXSY,5,0)                                                   BOUN   001C
C     SUBROUTINE BOUNDL                                                   BOUN   002I
      PROGRAM BOUNDL                                                      BOUN   003C
C     * * * * * * * * P  R  O  G  R  A  M    K  9  Q  A * * * * * * * *   BOUN   004
C                                                                         BOUN   005
C     SOLUTION OF THE 2-D , YAWED WING, AND AXISYMMETRIC, COMPRESSIBLE    BOUN   006
C     BOUNDARY LAYER EQUATIONS USING KELLER≠S BOX METHOD                  BOUN   007
C                                                                         BOUN   008
C     *****************************************************************  BOUN   009
C                                                                         BOUN   010
C     *****************     FORMULATED BY T. CEBECI     *****************  BOUN   011
C     *****************   AERODYNAMICS RESEARCH GROUP   **M**************  BOUN   012
C     ******  DOUGLAS AIRCRAFT DIVISION , MCDONNELL-DOUGLAS CORP. ******  BOUN   013
C     *****************     LONG BEACH , CALIFORNIA     *****************  BOUN   014
C                                                                         BOUN   015
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   BOUN   016
C                                                                         BOUN   017
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JI1,JIM1,NTC,NXT,NXW,NXM     BOUN   018
     1       ,TITLE(15)                                                   BOUN   019
      COMMON LG16,LG17,LG18,LG32,LG40                                     BOUN   020
     1       , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR       BOUN   021
      COMMON /HEADR/ CASE, IPAGE                                          BOUN   022
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)         BOUN   023
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)                  BOUN   024
      COMMON/BL11/VWPRI,UWPRI,DELV1                                       BOUN   025
      COMMON/BL13/FPSLN                                                   BOUN   026
      COMMON /BL16/ NTYPE,IOUT                                            BOUN   027
      COMMON/BLC5/EM(100,2),EDV(100,2),E(100,2),EB(100,2),VPRT(100)       BOUN   028
      COMMON/BLC7/VGP ,DETA1                                              BOUN   029
      COMMON/BLC8/A(100),CEL(100)                                         BOUN   030
      COMMON/BL10/SWP,TANL,WE,W(100,2),WP(100,2)                          BOUN   031
      COMMON/BL12/TI,RMI,UI,RI,PR,PRT,FK,RL,RMUI,RHOI,PSI,HE              BOUN   032
     1             ,UE(100),RO(100),TW(100),QW(100),RP(100),FW(100)       BOUN   033
     2             ,BR(100),TE(100),RHOE(100),RMUE(100),GW(100),GPW(100)  BOUN   034
     3             ,RF1(100),RF2(100),YS (100),IGX1(100),FPW(100),ROL(100) BOUN  035
```

```
      COMMON /BL14/ RX1,RTH1, CF0,CF1,CF2,CFSUMO,CFSUM,
     1        THETA(100),DELS(100),FPPW(100)
      COMMON /BL15/ NXY
      COMMON /BL17/ RX(100),CFA(100),CF(100),ETAE(100),CDI(100),ST(100),
     1         INP(100)
      COMMON/BL19/C(100,2),G(100,2),GP(100,2),
     1             RHO(100),RMU(100),TVCT(100)
      COMMON/BL20/ RTHTR, UEIN,ROIN,GAMAT
      COMMON/BL21/ A1(100,2),A2(100,2)
      COMMON/RADIUS/ ROMAX
C
C     - - - - - - - - - - - - - - - * - - - - - - - - - - - - - - -
C
      REWIND 2
      REWIND 3
      NTC=0
      IPAGE=1
   50 NTC=NTC+1
      IGNP=0
      IGTR=0
      NX=0
      IT = 0
      LSP=0
      IOUT = 0
C ----
      CALL INPT
      LC=0
      LCMAX=61
C ----
  100 NX=NX+1
C ----
      ITC=0
      IGRC=0
  120 IGOL = 0
      IGOT = 0
      IF(NX .LT. NXT)  IGOL = 1
      IF(NX .GE. NXT)  IGOT = 1
C ----
      IGCV=0
      CALL FINF
      IF(LG32 .NE. 1  .OR.  NX .EQ. 1)  CALL HEAD
      IF(LG32.EQ.1)  GO TO 130
      WRITE(6,6015) NX,BETA(NX),XI(NX),XS (NX),ETAINF(NX)
      GO TO 138
  130 IF(NX .EQ. 1) WRITE(6,7000)
      LC = LC+3
      IF(LC .LT. LCMAX)  GO TO 135
      CALL HEAD
      LC=0
  135 WRITE(6,7015) NX,BETA(NX),XI(NX),XS (NX),ETAINF(NX)
      LC = LC+2
  138 IF(LSP.EQ.1) GO TO 700
C ----
      IF(NX.EQ.1)  CALL IVPF
      IF(LSP.EQ.1) GO TO 700
C ----
      IF(NX.EQ.1)  CALL FLPR
      IF(LSP .EQ. 1)  GO TO 700
C ----
      IF( NX.NE.1 .OR. XS(NX).EQ.0.)  GO TO 140
      CALL EDVS
      IF(LSP.EQ.1) GO TO 700
C ----
  140 IF(NX.EQ.1)  GO TO 145
      CALL SHFT
      IF(LSP.EQ.1) GO TO 700
      CALL FLPR
      IF(LSP .EQ. 1)  GO TO 700
C----
  145 IT=0
```

```
      LC = LC+2                                                         BOUN  106
      IF(NX-NXT)150,142,150                                             BOUN  107
  142 ITC=1                                                             BOUN  108
  150 IT=IT+1                                                           BOUN  109
      LC = LC+1                                                         BOUN  110
      IF(IT.LE.9) GO TO 220                                             BOUN  111
      IF(ITC .EQ. 0) GO TO 200                                          BOUN  112
      WRITF(6,6000)                                                     BOUN  113
      ITC=0                                                             BOUN  114
      IT=1                                                              BOUN  115
      GO TO 220                                                         BOUN  116
  200 WRITE(6,6010)                                                     BOUN  117
      LSP=1                                                             BOUN  118
      GO TO 700                                                         BOUN  119
  220 CALL EDVS                                                         BOUN  120
      IF(LSP.EQ.1) GO TO 700                                            BOUN  121
      CALL MOMX                                                         BOUN  122
  300 IF(IGOL.EQ.1) GO TO 540                                           BOUN  123
      IF(IGOT.EQ.1) GO TO 550                                           BOUN  124
  540 IF( ABS(DELV1  ).LT.EPSLN) GO TO 600                              BOUN  125
      IF(V(1,2).LT.0. .AND. LG16.NE.0) GO TO 670                        BOUN  126
      GO TO 150                                                         BOUN  127
  550 EAG= DELV1/((V(1,2)+VWPRI)*.5)                                    BOUN  128
      IF( ABS(EAG).LT.0.02 ) GO TO 600                                  BOUN  129
      IF(IGTR .LE. 1 .OR. V(1,2) .LE. 0.)  GO TO 150                    BOUN  130
      IGCV=1                                                            BOUN  131
      CALL EINF                                                         BOUN  132
      IF(NX .EQ. 1)  GO TO 150                                          BOUN  133
      IF(LSP .EQ. 1)  GO TO 700                                         BOUN  134
      IF(IGRC .EQ. 0)  GO TO 150                                        BOUN  135
      CALL SHFT                                                         BOUN  136
      CALL FLPR                                                         BOUN  137
      GO TO 145                                                         BOUN  138
C----                                                                   BOUN  139
  600 WRITE(6,6030)    V(1,2)                                           BOUN  140
```

```
      LC = LC+1
      IF(IGTR.GT.1) IGTR=0
      IGCV=1
      CALL EINF
      LC=LC+3
      IF(LSP.EQ.1) GO TO 700
      IF(IGRC.EQ.0) GO TO 670
      CALL SHFT
      CALL FLPR
      IF(LSP .EQ. 1)  GO TO 700
      LC = LC+2
      GO TO 145
C ----
  670 CALL OTPT
      IF(NX.EQ.NXM) GO TO 700
C ----
      IF( IGOT .EQ. 1)  GO TO 100
      IF(NX .GT. 1 .AND. LG16 .NE. 0) CALL TRNS
      IF(IGTR .GT. 1) IGRC=1
      IF(LSP.EQ.1) GO TO 700
      IF(IGTR.GT.1) GO TO 120
C ----
      GO TO 100
  700 IF(NX .EQ. 0) GO TO 800
      IOUT=1
      CALL OTPT
  800 IF(LSP .EQ. 1) WRITE(6,9999)
C----
 5010 FORMAT( I3 ,39X,I1)
 6000 FORMAT(1H ,20X,43HCONVFRGENCE IS SLOW - ITERATIONS CONTINUING,/)
 6010 FORMAT(1H ,15X,45H*** ITERATIONS EXCEED THE ALLOWABLF LIMIT *** /)
 6015 FORMAT(1H0,///5X,8HSTATION=,I3,3X,5HBETA=,E16.9,2X,3HXI=,E16.9,2X,
     1       2HS=,E16.9,2X,8HETAINF =,E16.9,////)
 6030 FORMAT(1H ,32X,E20.9)
 7000 FORMAT(1H ,45X,20HOUTPUT IN SHORT FORM, /)
 7015 FORMAT(1H0,5X,8HSTATION=,I3,3X,5HBETA=,E16.9,2X,3HXI=,E16.9,2X,2HS BOUN  176
     1=,    E16.9,2X,8HETAINF =,E16.9 )                                 BOUN  177
 9999 FORMAT(1H0/45X,37H********** CASE TERMINATED **********   // )   BOUN  178
C     RETURN                                                            BOUN  179I
 1000 CONTINUE                                                          BOUN  180C
      END                                                               BOUN  181
      SUBROUTINE INPT
C
C **  SUBROUTINE INPT
C     THIS SUBROUTINE PROCESSES ALL THE INPUT DATA TO THE PROGRAM
C
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JT1,JTM1,NTC,NXT,NXW,NXM
     1        ,TITLE(15)
      COMMON LG16,LG17,LG18,LG32,LG40
     1        , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR
      COMMON /HEADR/ CASE, IPAGE
      COMMON/BLC7/VGP ,DETA1
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BL12/TI,RMI,UI,RI,PR,PRT,FK,RL,RMUI,RHOI,PSI,HE
     1             ,UE(100),RO(100),TW(100),QW(100),RP(100),FW(100)
     2             ,RR(100),TE(100),RHOE(100),RMUE(100),GW(100),GPW(100)
     3             ,RF1(100),RF2(100),YS (100),IGX1(100),FPW(100),ROL(100)
      COMMON/BL13/EPSLN
      COMMON /BL15/ NXY
      COMMON/RADIUS/ ROMAX
      DIMENSION S(301) ,XR(301) ,DUEDX(100) ,PE(100)
      DATA DATA1/1.4/, DATA2/6035.0/
C
C     - - - - - - - - - - - - - - * - - - - - - - - - - - - - - -
C
      EPSLN = .005
      PR = .72
C---- READ CASE DATA
      READ(5,5005) TITLE,CASE
      READ(5,5010) NXT,LG16,LG17,LG18,LG32,LG26,LG40,LG41
      READ(5,5020) TI,RMI,UI,FK,RL,RI
      READ(5,5025) ROMAX,DETA1,VGP
      IF(LG40 .EQ. 0) GO TO 7000
      READ(5,1013) NXM
      READ(5,1014) (XS(I),I=1,NXM)
      READ(5,1014) (YS(I),I=1,NXM)
      READ(5,1014) (UE(I),I=1,NXM)
      GO TO 7001
 7000 CONTINUE
      READ(3) NXM
      READ(3) (XS(I),I=1,NXM)
      READ(3) (YS(I),I=1,NXM)
      READ(3) (UE(I),I=1,NXM)
 7001 CONTINUE
      NXY=NXM
      DO 50 I=1,NXM
      RO(I) = YS(I)
      QW(I)=0.0
      RF2(I) = 0.
      FW(I) = 0.
      GW(I) = 0.
      GPW(I) = 0.
      RF1(I)=0.
      TW(I)=0.
      RP(I)=0.
      FPW(I)=0.
      BR(I)=0.
   50 IGX1(I)=0.
      ETAINF(1) = 6.
      ETAINF(2) = 10.
      NXNS=NXM
   85 CALL HEAD
      WRITE(6,2050) TITLE ,CASE
      WRITE(6,2500) LG16,LG17,LG18,LG32,NXT
  150 XS1=0.0
      SD1=0.0
  160 DO 180 I=2,NXM
      SDA1=(XS(I)-XS(I-1))**2+(YS(I)-YS(I-1))**2
      SQDA1= SQRT(SDA1)
      SD2=SD1+ ABS(SQDA1)
      S(2*I+1) = SD2
      SD1=SD2
      IF(I.EQ.2)  S(3)=XS1
  180 CONTINUF
      WRITF(2) NXM
      WRITF(2) (S(2*I+1),I=1,NXM)
      LC = 1
      LCMAX = 36
      WRITE(6,2550)
      DO 185 I=1,NXM
      IF(LC .LT. LCMAX) GO TO 182
      CALL HEAD
      WRITF(6,2550)
      LC = 1
      LCMAX = 49
  182 XBG = XS(I) * RL
      SBG = S(2*I+1) * RL
      WRITE(6,3100) I,XS(I),YS(I), XBG,SBG,S(2*I+1)
      LC = LC+1
      IF(FK .EQ. 1)  RO(I) = YS(I)*RL
      YS(I)=XS(I)
      XS(I) = SBG
  185 CONTINUE
      IF(LCMAX.EQ.36 .AND. LC.GT.18)  CALL HEAD
      IF(LCMAX.EQ.49 .AND. LC.GT.45)  CALL HEAD
      IF(FK .EQ. 0. .OR. LG18 .NE. 1) GO TO 203
      CALL SLOPE(NXM,XS,YS,RF1,1)
      IF(RL .EQ. 1.0)  GO TO 203
      DO 202 I=1,NXM
  202 RF1(I) = RF1(I)*RL
C----
  203 IF(UI.NE.0. .OR. RMI.NE.0.)  GO TO 204
      WRITE(6,9030)
      LSP = 1
      GO TO 1800
  204 IF(TT.NE.0.)  GO TO 205
```

```
      IF(LG41 .EQ. 0) GO TO 201
      WRITE(6,9045)
      TI=519. *(5./9.)
      GO TO 205
  201 CONTINUE
      WRITE(6,9040)
      TI = 519.0
  205 IF(RMI .NE. 0.)  UI = 0.
      IF(LG41 .EQ. 1) GO TO 206
      IF(UI.NE.0.) RMI=UI/( SQRT(TI)*49.1)
      IF(RMI.NE.0.) UI=RMI* SQRT(TI)*49.1
      RMUI= 1.0E-06*(.90311226E-03*TI+1.238522-(.56843634E-06*TI*TI
     1     +.38312556E-03*TI+1.436156)**0.5)
      RHOI=RMUI*RI/UI
      PSI=RHOI*TI*1718.0
      RMI2=RMI*RMI
      DK1=RMI2*(DATA1-1.0)*0.5
      UEA4=DATA1*RMI2*0.5
      SHI=DATA2*TI
  207 HE = SHI + .5*(UI**2 )
      GO TO 209
  206 TIR=TI*9./5.
      UII=UI/.3048
      IF(UI .NE. 0.) RMI=UII/(SQRT(TIR)*49.1)
      IF(RMI .NE. 0.) UI=(RMI*SQRT(TIR)*49.1)*.3048
      RMUI=1.0E-06*(.90311226E-03*TIR+1.238522-(.56843634E-06*TIR*TIR
     1+.38312556E-03*TIR+1.436156)**.5)
      RHOI=RMUI*RI/UII
      PSI=RHOI*TIR*1718.0
      SHI=DATA2*TIR
      HF=SHI+.5*(UII**2)
      RMUI=RMUI*47.88025
      RHOI=RHOI*515.379
      PSI=PSI*47.88025
      HE=HF*.3048*.3048
  209 CONTINUE
      IF(LG26.FQ.2) GO TO 270
      IF(LG26.EQ.3) GO TO 210
      WRITF(6,9050)
      LSP=1
      GO TO 1800
  210 CONTINUE
  208 DO 220 I=1,NXM
      UE(I)=UI* SQRT(1.0-UF(I))
  220 CONTINUE
      GO TO 300
  270 DO 280 I=1,NXM
      UE(I)=UE(I)*UI
  280 CONTINUE
C ----
  300 CONTINUE
      DO 320 I=1,NXM
      RHOE(I)=RHOI
      RMUE(I)=RMUI
      PE(I) = 0.
      TE(I) = 0.
  320 CONTINUE
  500 IST = 1
      MULT = 0
      CALL SLOPE(NXM,XS,UE,DUEDX,MULT)
C----
  510 DO 1500 I=1,NXM
      IF(I .GT. 1) GO TO 520
      VT1=RHOI*RMUI
  520 VT2=VT1*UE(I)
      VT3=VT2*UE(I)
      IF(FK.NE.0.) GO TO 550
      VT4=1.0
      VT5=1.0
      VT42=1.0
      GO TO 600
  550 IF(FK.NE.1.0) GO TO 560
      IF(RO(I) .NE. 0.)  GO TO 555
      IF(I .GT. 1)  WRITE(6,9095) I
      RO(I) = .0001*RL
  555 VT4=RO(I)/RL
      VT42=VT4*VT4
      VT5=RL/RO(I)
      GO TO 600
  560 WRITE(6,9060)
      LSP=1
      GO TO 1800
  600 ROL(I) = VT4
      FX2=VT2*VT42
      IF(I.EQ.1) XI(1)=FX2*XS(1)
      IF(I .EQ. 1)  GO TO 680





  650 XI(I)=XI(I-1)+(FX1+FX2)*(XS(I)-XS(I-1))*0.5
  680 FX1 = FX2
      IF(I .NE. 1)  GO TO 930
      IF(IST .EQ. 0)  GO TO 1500
      BETA(1) = 1.0
      IF(FK .EQ. 1.)  BETA(1) = BETA(1)/2.
      GO TO 1500
  930 IF(I .EQ. 2)  GO TO 950
      BETA(I)=2.*XI(I)/(VT3*VT42)*DUEDX(I)
      GO TO 1500
  950 BETA(I)=2.0*XI( I )*(UE(I)-UE(I-1))/(VT3*VT42*(XS(I)-XS(I-1)))
 1500 CONTINUE
C----
      WRITE(6,2600) DETA1, RL, PR, VGP, RHOI, SWP, FK,RMUI, HE, FPSLN
      1    ,UI, TI, RI, RMI
      CAIL HEAD
      WRITF(6,2900)
      WRITE(6,3000)
      LC=1
      LCMAX=45
      DO 1550 I=1,NXM
      IF(LC .LT. LCMAX) GO TO 1520
      CALL HEAD
      WRITE(6,3000)
      LC=1
      LCMAX=49
 1520 IF(FK .NE. 0.)  GO TO 1530
      WRITE(6,3200) I,YS(I),RO(I), TW(I), UE(I), PE(I), BR(I)
      GO TO 1540
 1530 WRITF(6,3200) I,YS(I),ROL(I),TW(I), UE(I), PF(I), BR(I)
 1540 CP=1.-UE(I)*UE(I)/(UI*UI)
      WRITF(6,3250) XS(I), RF1(I), QW(I),  CP  , RMUE(I), FPW(I)
      RMACH = 0.
      WRITE(6,3250) BETA(I), RF2(I), RP(I), RMACH, TE(I), XI(I)
      LC = LC+4
 1550 CONTINUE
 1590 IGXI=0
      IF(XI(1).GE.0.) GO TO 1600
      WRITF(6,9070)
      IGXI=1
 1600 IF(NXM .FQ. 1)  RETURN
      DO 1700 I=2,NXM
      IF(XI(I).GT.0.) GO TO 1620
      WRITE(6,9080) I
      IGXI=1
 1620 IF(XI(I).GT.XI(I-1)) GO TO 1700
      WRITF(6,9090) I
      IGXI=1
 1700 CONTINUE
```

```
      IF(IGXI.EQ.0) RETURN
      LSP=1
 1800 RETURN
C - - - -
 1013 FORMAT(I4)
 1014 FORMAT(6F10.0)
 2050 FORMAT(1H ,25X,15A4,10X,6HCASE ,A4,///1H ,54X,10H CASE DATA ,///)
 2500 FORMAT(1H0,10X,8HTRFLAG =,I1,10X,7HTRINT =,I1,10X,5HTVC =,I1,
     1 10X,8HSHORTP =,I1,//1H ,30X,31HTRANSITION SPECIFIED AT STATION,I4
     1)
 2550 FORMAT( 1H , 50X, 19HBODY GEOMETRY DATA  /
     1     1H0,21X,1HK,9X,3HX/C ,15X,3HY/C ,16X,1HX,17X,1HS ,16X,3HS/C /)
 2600 FORMAT(1H0/1H0,40X,43HREFERENCE QUANTITIES AND CONTROL PARAMETERS,
     A       /1H0,16X, 6HH1    = ,F9.5, 16X,8HC        = ,E15.7,10X,
     1     8HPRO    = ,F9.5, / 1H0,16X, 6HK     = ,F9.5, 16X,8HRHOREF = ,
     2     E15.7, 10X,8HSWEEP = ,F9.4 / 1H0,16X, 6HKK    = ,F9.5, 16X ,
     3     8HMUREF  = ,E15.7, 10X,8HE      = ,E15.7 / 1H0,16X, 6HEPS1 = ,
     4     F15.7, 10X, 8HVREF    = ,E15.7, 10X, 8HTREF   = ,F10.3/ 1H0,16X,
     5     31X,           8HREY     = ,E15.7, 10X,8HMREF   = ,E15.7/         )
 2900 FORMAT(1H ,50X,12HSTATION DATA//)
 3000 FORMAT(1H0,7X,1HN,12X,3HX/C,13X,4HRO/C,14X,2HTW,16X,2HUE,15X,2HPE,
     1       14X,2HFW,/1H ,20X,4H  S ,12X,6HALPHA1,13X,2HQW,16X,2HCP,14X,
     2       3HMUE,13X,3HFPW,/1H ,20X,4HBETA,12X,6HALPHA2,13X,2HRR,16X,
     3       2HME,15X,2HTE,12X,5HSQUIG,/ )
 3100 FORMAT(1H ,19X,I3,5(3X,E15.7 ))
 3200 FORMAT(1H /1H ,I8,3X,6E17.6)
 3250 FORMAT(1H ,11X, 6E17.6)
 5005 FORMAT(15A4,A4)
 5010 FORMAT(I4,7I1)
 5020 FORMAT(5F10.0,E12.6)
 5025 FORMAT(3F10.0)
 5030 FORMAT(2F8.0,F6.0,F5.0,F6.0,F7.0,2F6.0,F7.0,8X,F4.0,I1 )
 5040 FORMAT(3E14.9)
 9004 FORMAT(1H ,37H**ERROR - INPUT REFERENCE LENGTH = 0. /)
 9005 FORMAT(1H ,51H**ERROR - INPUT SURFACE DISTANCE AT STATION 1 LT 0.)
```

```
9006 FORMAT(1H ,66H**ERROR - INPUT SURFACE DISTANCE NOT IN ASCENDING OR
    1DER AT STATION, I3 //)
9010 FORMAT(1H ,51H**ERROR - INPUT X NOT IN ASCENDING ORDER AT STATION,
    1        I3 //)
9020 FORMAT(1H0,51H**ERROR - INPUT NTR OR S AT STATION 1 ARE INCORRECT)
9030 FORMAT(1H0,42H**ERROR - NO INPUT FOR EITHER VREF OR MREF)
9040 FORMAT(1H0,52H***** WARNING - TREF INPUT = 0., VALUE RESET TO 519.
    1)
9045 FORMAT(1H0,67H**** WARNING - TREF INPUT = 0., VALUE RESET TO 288.3
    133 DEGREES KELVIN     )
9050 FORMAT(1H0,27H**ERROR - CHECK CPCOM INPUT )
9060 FORMAT(1H0,48H**ERROR - INPUT FLOW INDEX NOT EQUAL TO 1. OR 0. )
9070 FORMAT(1H ,38H**ERROR - XI AT STATION 1 NE OR GT  0. )
9080 FORMAT(1H ,25H** ERROR - XI AT STATION ,I3,12H IS NEGATIVE)
9090 FORMAT(1H ,25H** ERROR - XI AT STATION ,I3,26H IS NOT IN ASCENDING
    1 ORDER)
9095 FORMAT(1H0,41H****** WARNING - ROIC INPUT=0. AT STATION , I3,
    1     23H - VALUE RESET TO .0001 )
9100 FORMAT(1H0, 40X,25HINSS NOT SUCCESSFUL NER = ,I2,2X,10HAT STATION,
    1       1X,I3/1H ,12X,1HI,15X,3HX/C,22X,3HY/C, //)
9150 FORMAT(1H0,47H** ERROR - INPUT CP EXCEEDS ALLOWABLE LIMITS OF,
    1       2E17.6,1X, 10HAT STATION,I3 /)
9200 FORMAT(1H ,11X,I3,2E20.9)
9300 FORMAT(1H0,40X,25HINSR NOT SUCCESSFUL NER = ,I2,2X,10HAT STATION,
    1       1X,I3,/1H ,12X,1HI,15X,3HXIC,22X,3HYIC, //)
     END
```

```
      SUBROUTINE EINF
C **  SUBROUTINE EINF
C     THIS SUBROUTINE CALCULATES THE TRANSFORMED Y - GRID POINTS
C
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JT1,JTM1,NTC,NXT,NXW,NXM
     1      ,TITLE(15)
      COMMON LG16,LG17,LG18,LG32,LG40
     1      , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BLC7/VGP ,DETA1
C
C     - - - - - - - - - - - - - - - - * - - - - - - - - - - - - - - -
C
      IF(IGCV.EQ.1) GO TO 30
   30 IF(NX.EQ.1) GO TO 50
      IF(NX.EQ.2) GO TO 250
      IF(IGNP.EQ.1) GO TO 300
      IF(IGNP.EQ.0) GO TO 800
      WRITE(6,9910)
      LSP=1
      GO TO 1800
C ----
   50 IF(IGNP.EQ.0) GO TO 1000
      IF(IGNP.EQ.1) GO TO 100
      WRITE(6,9920)
      LSP=1
      GO TO 1800
C ----
  100 DO 120 J=2,NP
      IF( ABS(V(J,2)).LT.1.E-5) GO TO 500
      IF( ABS(V(J,2)).LT.1.E-6) GO TO 500
  120 CONTINUE
      DO 140 J=2, NP
      IF( ABS(V(J,2)).LT.1.E-4) GO TO 500
      IF(U(J,2).GE..999999) GO TO 500
      IF(U(J,2).LT.0.) GO TO 400
  140 CONTINUE
      WRITE(6,9930)
      J = NP
      GO TO 530
C ----
  250 IF(IGNP.EQ.0) GO TO 1000
  300 DO 320 J=JI,NP
      TF( ABS(V(J,2)).LT.1.E-5) GO TO 500
      IF( ABS(V(J,2)).LT.1.E-6) GO TO 500
  320 CONTINUE
      DO 340 J=JI,NP
      IF( ABS(V(J,2)).LT.1.E-4) GO TO 500
      IF(J .EQ.NP) GO TO 330
      IF(U(J,2).GE..999999) GO TO 500
  330 IF(U(J,2).LT.0.) GO TO 400
  340 CONTINUE
C ----
      WRITE(6,9980)
      IGRC=1
      ETAINF(NX)=ETAINF(NX)+10.
      GO TO 1010
C ----
  400 WRITE(6,9940) J
      LSP=1
      GO TO 1800
C ----
  500 IF(IGTR .GT. 1)  GO TO 600
  530 ETAINF(NX)=ETA(J)
      JI=J
      IGNP=0
      WRITE(6,6010) ETAINF(NX)
      GO TO 1800
  600 IGTR=0
```

```
      RETURN
C ----
  800 K = NX-1
      IF(K .EQ. NXT)  GO TO 820
      IF((ETAINF(NX-1).GT.ETAINF(NX-2)).AND.(IGTR .LE. 1)) GO TO 850
      ETAINF(NX) = ETAINF(NX-1) + 2.
  820 IF(IGOT.EQ.1) ETAINF(NX)=ETAINF(NX-1)+10.
      GO TO 1000
  850 ETAINF(NX)=ETAINF(NX-2)+(XI(NX)-XI(NX-2))/(XI(NX-1)-XI(NX-2))*
     1(ETAINF(NX-1)-ETAINF(NX-2))
C ----
 1000 IF(NX.GT.1) NPPR=NP
 1010 IF(VGP.EQ.1.)GO TO  1020
      ARGLOG=1.+ETAINF(NX)/DETA1*(VGP-1.)
      DLOG1=ALOG(ARGLOG)
      ARGINT=1.+DLOG1/ALOG(VGP)
      NP=  INT(ARGINT)+1
      GO TO 1050
 1020 ARGINT=ETAINF(NX)/DETA1+1.
      NP=  INT(ARGINT)
 1050 NPM1=NP-1
      IF(NP.LE.100) GO TO 1060
      WRITE(6,9970) NX, NP
      LSP=1
      GO TO 1800
 1060 ETA(1)=0.
      DELETA(1)=DETA1
      M = 1
      M1 = M+1
      NP = M*(NP-1)+1
      NPM1 = NP-1
      DO 1080  J= M1,NP,M
      N = J-M+1
      ETA(J) = DETA1 + VGP*ETA(N-1)
      DELETA(J-1) = ETA(J) - ETA(N-1)
      IF(M .EQ. 1)  GO TO 1080
      DELETA(J-1) = DELETA(J-1)/M
      ETA(J-1) = ETA(J) - DELETA(J-1)
      DELETA(J-2) = DELETA(J-1)
      IF(M .EQ. 2)  GO TO 1080
      ETA(J-2) = ETA(J-1) - DELETA(J-2)
      DELETA(J-3) = DELETA(J-1)
 1080 CONTINUE
      IF(VGP.NE.1.)ETAINF(NX)=ETA(NP)
      IGNP=1
 1800 RETURN
C ----
 6010 FORMAT(1H ,/16X,10H* ETAE  = ,F10.6)
 9910 FORMAT(1H ,22H** ERROR AT ETAE(1) **)
 9920 FORMAT(1H ,22H** ERROR AT ETAE(2) **)
 9930 FORMAT(1H0,49H** WARNING - INPUT ETAE AT STATION   IS TOO SMALL /
     1       1H ,48H** CALCULATIONS CONTINUING WITH THE INPUT ETAINF )
 9940 FORMAT(1H ,1X,39H**ERROR - FP PROFILE IS NEGATIVE AT I =, I3)
 9970 FORMAT(1H ,1X,16H**ERROR - IMAX (, I3, 20H) EXCEEDS 100 -IMAX=,I3)
 9980 FORMAT(1H ,38H** WARNING - ETAE IS BEING REESTIMATED )
C2000 CONTINUE
      END
      SUBROUTINE IVPF
C
C **  SUBROUTINE IVPF
C     THIS SUBROUTINE GENERATES THE INITIAL VELOCITY PROFILE
C
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JI1,JIM1,NTC,NXT,NXW,NXM
     1       ,TITLE(15)
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BLC5/EM(100,2),EDV(100,2),E(100,2),EB(100,2),VPRT(100)
C     - - - - - - - - - - - - - - * - - - - - - - - - - - - - - -
C
      IF(IT .LE. 1 .AND. NX.EQ.1)  E(1,1) = 1.
      F(1,2)=0.
      U(1,2)=0.
C ----POLHAUSEN INITIAL VELOCITY PROFILE
      VPGT = ETAINF(1)*BETA(1)/6.
      V(1,2)=2./ETAINF(1) + VPGT
      DO 50 J=2,NP
      EDVE=ETA(J)/ETAINF(1)
      EDVE2=EDVE*EDVE
      EDVE3=EDVE2*EDVE
      EDVE4=EDVE3*EDVE
      FPGT = (.5-EDVE+.75*EDVE2-.2*EDVE3)*ETA(J)**2*ETAINF(1)*BETA(1)/6.
      F(J,2)=EDVE2*ETAINF(1)*(1.-0.5*EDVE2+0.2*EDVE3) + FPGT
      UPGT = EDVE*(1.-3.*(EDVE-EDVE2)-EDVE3)*ETAINF(1)**2*BETA(1)/6.
      U(J,2)=2.*EDVE-2.*EDVE3+EDVE4 + UPGT
      VPGT = (1.-6.*EDVE+9.*EDVE2-4.*EDVE3)*ETAINF(1)*BETA(1)/6.
      V(J,2)=2./ETAINF(1)*(1.-3.*EDVE2+2.*EDVE3) + VPGT
   50 CONTINUE
 1800 RETURN
      END
      SUBROUTINE FLPR
C
C * * SUBROUTINE FLPR
C     CALCULATE THE FLUID PROPERTIES AND THE TVC TERM
C
      COMMON   NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JI1,JIM1,NTC,NXT,NXW,NXM
     1      ,TITLE(15)
      COMMON LG16,LG17,LG18,LG32,LG40
     1      , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BL12/TI,RMI,UI,RI,PR,PRT,FK,RL,RMUI,RHOI,PSI,HE
     1           ,UE(100),RO(100),TW(100),QW(100),RP(100),FW(100)
     2           ,BR(100),TE(100),RHOE(100),RMUE(100),GW(100),GPW(100)
     3           ,RF1(100),RF2(100),YS (100),IGX1(100),FPW(100),ROL(100)
      COMMON/BL19/C(100,2),G(100,2),GP(100,2),
     1            RHO(100),RMU(100),TVCT(100)
C
C     - - - - - - - - - - - - - - * - - - - - - - - - - - - - - - - -
C
      A1 = 0.
      IF(XI(NX) .GT. 0.) A1 = SQRT(2.*XI(NX))/(  RHOI  *UE(NX))
C ----
      G(NP,2) = 1.0
      SUM = 0.
      F1 =  RF1(NX)
      TVCT(1) = 1.0
      DO 500 I=2,NP
      IF(LG18 .EQ. 0)  GO TO 450
      F2 =  RF1(NX)
      SUM = SUM + (F1+F2)/2.*DELETA(I-1)
      TVCT(I) = SQRT(1. + 2.*A1*SUM/(RL*ROL(NX)*ROL(NX)))
      F1 = F2
      GO TO 500
C---- IF NO TVC THEN TVC - TERM = 1.0
```

```
  450 TVCT(I) = 1.0                                                     FLPR  036
  500 CONTINUE                                                          FLPR  037
 1800 RETURN                                                            FLPR  038
      END                                                               FLPR  039
      SUBROUTINE EDVS
C
C  **  SUBROUTINE FDVS
C      THIS SUBROUTINE COMPUTES THE EDDY VISCOSITY
C
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JI1,JIM1,NTC,NXT,NXW,NXM
     1      ,TITLE(15)
      COMMON LG16,LG17,LG18,LG32,LG40
     1      , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BLC5/FM(100,2),EDV(100,2),E(100,2),FB(100,2),VPRT(100)
      COMMON/BL10/SWP,TANL,WE,W(100,2),WP(100,2)
      COMMON/BL12/TI,RMI,UI,RI,PR,PRT,FK,RL,RMUI,RHOI,PSI,HE
     1           ,UE(100),RO(100),TW(100),QW(100),RP(100),FW(100)
     2           ,BR(100),TE(100),RHOE(100),RMUE(100),GW(100),GPW(100)
     3           ,RF1(100),RF2(100),YS (100),IGX1(100),FPW(100),ROL(100)
      COMMON/BL19/C(100,2),G(100,2),GP(100,2),
     1            RHO(100),RMU(100),TVCT(100)
      COMMON/BL20/ RTHTR, UEIN,ROIN,GAMAT
      DIMENSION  EDVI(100), EDVO(100) ,EDVM(100)
      DATA  DATA1/.0168/,DATA2/.40 /,DATA3/.080/,DATA4/.44/
C
C     - - - - - - - - - - * - - - - - - - - - -
C
      IF(IGOL.EQ.1) GO TO 500
      SQ2XI = SQRT(2.*XI(NX))
      TC = RHOI     *UE(NX)*ROL(NX)/SQ2XI
      IF(IGOT.EQ.1) GO TO 600
C---- E P S. F O R  L A M I N A R  F L O W S
  500 DO 520 J = 1, NP
      EM(J,2)= 1.0
      EDV(J,2) = 0.
      VPRT(J) = PRT
  520 CONTINUE
```

```
      GO TO 3000
C---- E P S   F O R   T U R B U L E N T   F L O W S
  600 SUM = 0.
      SUMT = 0.
      F1 = 1.0
      F3 = 0.
      DO  620 J=1,NP
      EM(J,2)=1.0
      EDV(J,2) = 0.
      IF(J .EQ. 1)  GO TO 620
      F2 = (1.-U(J,2))/TVCT(J)
      F4 = F2*U(J,2)
      SUM = SUM + (F1+F2)/2.*DELETA(J-1)
      SUMT = SUMT + (F3+F4)/2.*DELETA(J-1)
      F1 = F2
      F3 = F4
  620 CONTINUE
      RTHI = UE(NX)*(RHOI/RMUI     ) * SUMT/TC
      IF(RTHI .LT.  425.) GO TO 720
      IF(RTHI .GT. 6000.) GO TO 750
      XPHI= RTHI/425.-1.0
      CPHI= .55*(1.-EXP(-.243*SQRT(XPHI)-.298*XPHI))
      GO TO 770
  720 CPHI = 0.0
      GO TO 770
  750 CPHI=.55
  770 DATAN=DATA1*(1.55/(1.+CPHI))
C----
      IFLGED = 0
      IF(IT .LE. 1)  E(1,2) = E(1,1)
      J = 1
      SUM1 = 0.
      F1=1.0
      VWPSQT = V(1,2)
      DUDYW = TC*UE(NX)*ABS(VWPSQT)
      TAUW = RMUI     *DUDYW *E(1,2)                                    EDVS  071
      USTAR = SQRT(TAUW/RHOI)                                           EDVS  072
 1025 DPDX = -TC*TC*UE(NX)*RMUI     *BETA(NX)                           EDVS  073
      PPLUS = -RMUI     *DPDX/(RHOI*RHOI          *USTAR*USTAR* USTAR)  EDVS  074
      A = 26.                                                           EDVS  075
      ARMT = 1.-11.8*PPLUS                                              EDVS  076
 1030 CONTINUE                                                          EDVS  077
 1045 APLUS = A/SQRT(ARMT)                                              EDVS  078
      EDVO(J) = DATAN*UE(NX)*(RHOI/RMUI     ) * ABS(SUM/TC)             EDVS  079
      VPRT(J) = PRT                                                     EDVS  080
      IF (IFLGED .EQ. 1) GO TO 1098                                     EDVS  081
      F2 = 1./TVCT(J)                                                   EDVS  082
      IF(J .EQ. 1)  GO TO 1060                                          EDVS  083
      SUM1 = SUM1 + (F1+F2)/2.*DELETA(J-1)                              EDVS  084
      F1 = F2                                                           EDVS  085
 1060 Y = SUM1  /TC                                                     EDVS  086
      IF(TVCT(J) .GT. 1.005)  Y = RO(NX)*ALOG(TVCT(J))                  EDVS  087
      YPLUS = Y*USTAR*RHOI/RMUI                                         FDVS  088
      YA = YPLUS/APLUS                                                  EDVS  089
      EL = DATA2*Y                                                      EDVS  090
      IF(YA .LT. 20.)  EL = EL *(1.-EXP(-YA))                           EDVS  091
      BPLUS = 34.                                                       EDVS  092
      IF(J .NE. 1)  GO TO 1065                                          EDVS  093
      VPRT(J) = DATA2/DATA4 * BPLUS/APLUS                               EDVS  094
      GO TO 1070                                                        EDVS  095
 1065 VPRT(J) = EL/(DATA4*Y)                                            EDVS  096
      YB = YA*APLUS/BPLUS                                               EDVS  097
      IF(YB .LT. 20.)  VPRT(J) = VPRT(J)/(1.-EXP(-YB))                  EDVS  098
 1070               VWPSQT=V(J,2)                                       FDVS  099
      DUDY = TC*UE(NX)*ABS(VWPSQT) / F2                                 EDVS  100
      EDVI(J) = EL*EL*DUDY*RHOI/RMUI          * TVCT(J)                 EDVS  101
      IF(EDVI(J) .LT. EDVO(J))  GO TO 1100                              EDVS  102
      IFLGED=1                                                          EDVS  103
      IF (J .LE. 2)  WRITE(6,9030)                                      EDVS  104
 1098 EDV(J,2)=EDVO(J)                                                  EDVS  105
```

```
      GO TO 1200
 1100 EDV(J,2) = EDVI(J)
 1200 J = J + 1
      IF (J .LE. NP) GO TO 1030
C----
      IF(IFLGED.EQ.1) GO TO 2050
      WRITE(6,9020)
      EDV(J,2)=EDVO(J)
 2050 IF(LG17.EQ.0 .OR. IGTR.EQ.2)  GO TO 2500
      UR = UE(NX)*RHOI/RMUI
      IF(NX .GT. NXT)  GO TO 2150
      F1 = 0.
      SUMT = 0.
      DO 2100 J=2,NP
      F2 = U(J,2)*(1.-U(J,2))
      SUMT = SUMT + (F1+F2)*DELETA(J-1)*.5
      F1 = F2
 2100 CONTINUE
      RTHTR = UR*SUMT/TC
      IF(LG16 .NE. 0)  GO TO 2300
      UEIN = 0.
      ROIN = 0.
      GO TO 2300
 2150 IF(IT .GT. 1)  GO TO 2500
      UEIN = UEIN + .5*((1./UE(NX))+(1./UE(NX-1)))*(XS(NX)-XS(NX-1))
      IF(FK .EQ. 0.)  GO TO 2200
      ROIN = ROIN + .5*((1./RO(NX))+(1./RO(NX-1)))*(XS(NX)-XS(NX-1))
      GO TO 2300
 2200 ROIN = XS(NX)-XS(NXT)
 2300 ATR = 60.
      GTR = ((UR/ATR)**2)*UE(NX)/(RTHTR**2.68)
      AREXP = GTR*UEIN*ROIN
      IF(FK .NE. 0.)  AREXP = AREXP*RO(NXT)
      IF(AREXP .GT. 10.)  GO TO 2500
      GAMAT = 1. - EXP(-AREXP)
      WRITE(6,9500)  GAMAT
 2500 DO 2550 J=1,NP
      IF(J.GT.1) GO TO 2510
      EDVM(1) = EDV(1,2)
      GO TO 2550
 2510 IF(J.EQ.NP) GO TO 2520
      EDVM(J) =(EDV(J-1,2)+EDV(J,2)+EDV(J+1,2))/3.0
      GO TO 2550
 2520 EDVM(NP) =(EDV(NP-2,2)+EDV(NP-1,2)+EDV(NP,2))/3.0
 2550 CONTINUE
      DO 2560 J=1,NP
      EDV(J,2)=EDVM(J)
      IF(LG17.EQ.0 .OR. IGTR.EQ.2)  GO TO 2560
      EDV(J,2) = EDV(J,2)*GAMAT
 2560 CONTINUE
 3000 E(1,2)=(EM(1,2)+EDV(1,2))
      DO 3010 J=2,NP
      E(J,2)=(EM(J,2)+EDV(J,2))       *TVCT(J)*TVCT(J)
      EB(J-1,2) = 0.5* (E(J,2) + E(J-1,2))
 3010 CONTINUE
 1800 RETURN
C----
 9010 FORMAT(1H ,30X,43H**PPLUS EXCEEDS THE LAMINARIZATION LIMIT **)
 9020 FORMAT(1H ,30X,45H**NOTE - EPS DISTRIBUTION = EPS(INNER) ONLY**)
 9030 FORMAT(1H ,30X,45H**NOTE - EPS DISTRIBUTION = EPS(OUTER) ONLY**)
 9500 FORMAT(1H ,41X,11HGAMMA(TR) =, E17.6)
      END
      SUBROUTINE SHFT                                                   SHFT 001
C **  SUBROUTINE SHFT                                                   SHFT 002
C     THIS SUBROUTINE PROVIDES THE INITIAL GUESSES FOR EACH STATION     SHFT 003
C                                                                       SHFT 004
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JI1,JIM1,NTC,NXT,NXW,NXM   SHFT 005
     1      ,TITLE(15)                                                  SHFT 006
      COMMON LG16,LG17,LG18,LG32,LG40                                   SHFT 007
     1      , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR      SHFT 008
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)       SHFT 009
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)                SHFT 010
      COMMON/BLC5/EM(100,2),EDV(100,2),E(100,2),EB(100,2),VPRT(100)     SHFT 011
      COMMON/BL19/C(100,2),G(100,2),GP(100,2),                          SHFT 012
     1             RHO(100),RMU(100),TVCT(100)                          SHFT 013
      COMMON/BL21/ A1(100,2),A2(100,2)                                  SHFT 014
C                                                                       SHFT 015
C     - - - - - - - - - - - - - - - * - - - - - - - - - - - - - - -     SHFT 016
C                                                                       SHFT 017
      IF(IGRC.EQ.1) GO TO 200                                           SHFT 018
                  JI1=JI+1                                              SHFT 019
   50 JIM1=JI-1                                                         SHFT 020
      PHI=F(JI,2)-ETAINF(NX-1)                                          SHFT 021
      DO 70 J=1,JI                                                      SHFT 022
   60 F(J,1)=F(J,2)                                                     SHFT 023
      U(J,1)=U(J,2)                                                     SHFT 024
      V(J,1)=V(J,2)                                                     SHFT 025
   65 EDV(J,1)=EDV(J,2)                                                 SHFT 026
      E(J,1)=E(J,2)                                                     SHFT 027
      IF(J.EQ.JI) GO TO 70                                              SHFT 028
      EB(J,1)=EB(J,2)                                                   SHFT 029
   70 CONTINUE                                                          SHFT 030
   80 DO 90 J=JI1,NP                                                    SHFT 031
   85 F(J,1)=PHI+ETA(J)                                                 SHFT 032
      U(J,1)=1.                                                         SHFT 033
      V(J,1)=0.                                                         SHFT 034
      EDV(J,1)=EDV(JI,2)                                                SHFT 035
      E(J,1)=E(JI,2)
      FB(J-1,1)=EB(JIM1,2)
   90 CONTINUE
      IF(IGRC.EQ.0) GO TO 100
      IGRC=0
      GO TO 1800
C ----
  100 DO 120 J=JI1,NP
      F(J,2)=PHI+ETA(J)
      U(J,2)=1.
      V(J,2)=0.
  120 CONTINUE
      IF(IGRC.EQ.1) GO TO 80
      GO TO 1800
  200 DO 220 J=1,JI
  210 F(J,2)=F(J,1)
      U(J,2)=U(J,1)
      V(J,2)=V(J,1)
  215 EDV(J,2)=EDV(J,1)
      F(J,2)=E(J,1)
      IF(J.EQ.JI) GO TO 220
      EB(J,2)=EB(J,1)
  220 CONTINUE
      PHI=F(JI,2)-ETAINF(NX-1)
      IF(IGTR .GT. 1)  PHI = F(JI,2)-ETAINF(NX)
      GO TO 100
 1800 RETURN
      END
      SUBROUTINE MOMX
C
C **  SUBROUTINE MOMX
C     FIND THE SOLUTION OF THE X-MOMENTUM EQUATION
C
      COMMON  NX,NP,NPPR,JT,IT,NRVP,LSP,NPM1,JT1,JIM1,NTC,NXT,NXW,NXM
     1      ,TITLE(15)
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)
      COMMON LG16,LG17,LG18,LG32,LG40
     1      , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BLC5/FM(100,2),EDV(100,2),E(100,2),EB(100,2),VPRT(100)
      COMMON/BLC8/A(100),CEL(100)
      COMMON/BL11/VWPRI,UWPRI,DELV1
      COMMON/BL12/TI,RMI,UI,RI,PR,PRT,FK,RL,RMUI,RHOI,PSI,HE
     1           ,UF(100),RO(100),TW(100),QW(100),RP(100),FW(100)
     2           ,RR(100),TE(100),RHOE(100),RMUE(100),GW(100),GPW(100)
     3           ,RF1(100),RF2(100),YS (100),IGX1(100),FPW(100),ROL(100)
      COMMON/BL19/C(100,2),G(100,2),GP(100,2),
     1            RHO(100),RMU(100),TVCT(100)
      DIMENSION A1(100),B1(100),G1(100),D(100),SF(100),S(100),
     1                X(100),Y(100),Z(100),DELF(100),DELU(100),DELV(100)
C
C     -------------------- * --------------------
C
      IF(IT.GT.1) GO TO 100
      IF(NX.EQ.1) CEL(1)=0.
      IF(NX.GT.1) CEL(NX)=2.*XI(NX-1)/(XI(NX)-XI(NX-1))+1.
      VWPRI=V(1,1)
      UWPRT=U(1,1)
C ----
  100 VWPRI=V(1,2)
      UWPRT=U(1,2)
C ----
      A(1) = 0.
      DO 320 J = 2, NP                                               MOMX  036
  320 A(J) = DELETA(J-1) / 2.                                        MOMX  037
C ----                                                               MOMX  038
      DO 900 J = 2, NP                                               MOMX  039
      FB      =(F(J,2) + F(J-1,2)) * 0.5                             MOMX  040
      VB      =(V(J,2) + V(J-1,2)) * 0.5                             MOMX  041
      IF (NX .GT. 1) GO TO 600                                       MOMX  042
      CFB     = 0.                                                   MOMX  043
      CUB     = 0.                                                   MOMX  044
      CVB     = 0.                                                   MOMX  045
      GO TO 700                                                      MOMX  046
  600 CFB      =(F(J,1) + F(J-1,1)) * 0.5                            MOMX  047
      CUB      =(U(J,1) + U(J-1,1)) * 0.5                            MOMX  048
      CVB      =(V(J,1) + V(J-1,1)) * 0.5                            MOMX  049
C ----                                                               MOMX  050
  700 TM1 = A(J) /FB(J-1,2)                                          MOMX  051
      A1(J) = TM1 * ((1. + CEL(NX)) * VB      + CEL(NX) * CVB     )  MOMX  052
      B1(J) = 1. + TM1 * ((E  (J,2)-E  (J-1,2))/DELETA(J-1) + (1.    MOMX  053
     1       + CEL(NX)) * FB      - CEL(NX) * CFB     )              MOMX  054
C ----                                                               MOMX  055
      IF (NX .EQ. 1) RB     = 0.                                     MOMX  056
      IF (NX .GT. 1) RB   =-(EB(J-1,1) * ((V(J,1) - V(J-1,1)) / DELETA MOMX  057
     1     (J-1)) + (((E  (J,1) - E  (J-1,1)) / DELETA(J-1)) + CFB   MOMX  058
     2   )*CVB +BETA(NX-1)*.5                -CUB*CUB*BETA(NX-1)      MOMX  059
     3  -CEL(NX)*(CFB*CVB-CUB*CUB) + BETA(NX-1)*.5)                   MOMX  060
      S(J)=V(J-1,2)-V(J,2)-DELETA(J-1)/EB(J-1,2)        * ((1. + CEL(NX) MOMX  061
     1    ) *FB*VB    -  (BETA(NX)+CEL(NX))*((U(J,2)+U(J-1,2))*.5)**2 MOMX  062
     2   -CEL(NX)*CFB*VB + BETA(NX)*.5              +CEL(NX)*CVB*FB -RB MOMX  063
     3 +VB*(E(J,2)-E(J-1,2))/DELETA(J-1) + BETA(NX)*.5)              MOMX  064
  900 CONTINUE                                                       MOMX  065
C ----                                                               MOMX  066
  905 D(2) = -.5*(-A(2)/EB(1,2)*(BETA(NX)+CEL(NX))*(U(2,2)+U(1,2))   MOMX  067
     1    +A(2)*A1(2) + B1(2)/A(2))                                  MOMX  068
      SE(2)= - A(2)                                                  MOMX  069
      G1(2) = - D(2) - 1. / A(2)                                     MOMX  070
      Y(2) = F(1,2)-F(2,2) + DELFTA(1)*.5*(U(2,2)+U(1,2))
      X(2) =-0.5 * (S(2) + A1(2) * Y(2) + B1(2) * (U(1,2)-U(2,2)+
     1   DELETA(1)*(V(2,2)+V(1,2))*.5)/A(2))
      Z(2) = -X(2)-(A(2)*(V(2,2)+V(1,2)) +U(1,2)-U(2,2))       /A(2)
      DO 950 J = 3, NP
      TMR11 = -A(J) +SE(J-1)
      TMR21 = - A1(J) *SE(J-1) + G1(J-1) * (2. - B1(J))-(U(J,2)+U(J-1,2)
     1   )*A(J)/EB(J-1,2)*(BETA(NX)+CEL(NX))
      TMR31 = -1. + A(J) * G1(J-1)
      D(J) = (A(J) * A(J) * A1(J) -(A(J)**2)*(BETA(NX)+CEL(NX)) *
     1   (U(J,2)+U(J-1,2))/EB(J-1,2) + B1(J))
     2      / (-TMR11 * A1(J) * A(J) + A(J) * TMR21 + TMR31 * B1(J))
      SE(J)= - A(J) - TMR11 * D(J)
      G1(J) = (TMR31 * D(J) - 1.) / A(J)
      TMM1 = A(J)*(U(J-1,2)+U(J,2))+F(J-1,2)-F(J,2) + Y(J-1)
      TMM2 = S(J) - A1(J) * Y(J-1) - (B1(J) - 2.) * Z(J-1)
      TMM3 = A(J)*(V(J,2)+Z(J-1) +V(J-1,2)) - U(J,2)+U(J-1,2)
      DNTR = - A(J) * TMR11 * A1(J) + A(J) * TMR21 + TMR31 * B1(J)
      X(J) = (- A(J) * A1(J) * TMM1 + A(J) * TMM2 + B1(J) * TMM3) / DNTR
      Y(J) = TMM1 - TMR11 * X(J)
      Z(J) = (TMR31 * X(J) - TMM3) / A(J)
  950 CONTINUE
C ----
      DELF(NP) = Y(NP)
      DELU(NP-1) = X(NP)
      DELV(NP) = Z(NP)
C ----
      J = NP
 1000 J = J - 1
      DELF(J) = Y(J) -SE(J) * DELU(J)
      DELU(J-1) = X(J) - D(J) * DELU(J)
      DELV(J) = Z(J) - G1(J) * DELU(J)
      IF (J .GT. 3) GO TO 1000
      DELF(2) = Y(2) -SE(2) * DELU(2)
      DELV(2) = Z(2) - G1(2) * DELU(2)
      DELF(1) = 0.
      DELV(1) = X(2) - D(2) * DELU(2)
      DELU(1) = 0.
C ----
      IF (IT.EQ. 1) WRITE (6, 9510)
      WRITE (6, 9521)IT, V(1,2), DELV(1)
C ----
      DO 1020 J=1,NP
      IF (J .EQ. NP) GO TO 1010
      U(J,2) = U(J,2) + DELU(J)
 1010 F(J,2) = F(J,2) + DELF(J)
      V(J,2) = V(J,2) + DELV(J)
 1020 CONTINUE
      DELV1 = DELV(1)
 1800 RETURN
C ----
 9510 FORMAT (1H0,  21X,1HI,20X,4HFPPW ,  26X,5HDELVW      )
 9521 FORMAT(1H ,20X,I2,10X,E20.9,10X,E20.9)
      END
      SUBROUTINE TRNS
C
C **  SUBROUTINE TRNS
C     THIS SUBROUTINE COMPUTES THE LOCATION OF B. L. TRANSITION
C
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JI1,JIM1,NTC,NXT,NXW,NXM
     1       ,TITLE(15)
      COMMON LG16,LG17,LG18,LG32,LG40
     1       , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BL12/TI,RMI,UT,RI,PR,PRT,FK,RL,RMUI,RHOI,PSI,HE
     1             ,UE(100),RO(100),TW(100),QW(100),RP(100),FW(100)
     2             ,RR(100),TE(100),RHOE(100),RMUE(100),GW(100),GPW(100)
     3             ,RF1(100),RF2(100),YS (100),IGX1(100),FPW(100),ROL(100)
      COMMON /BL14/ RX1,RTH1, CF0,CF1,CF2,CFSUM0,CFSUM,
     1        THETA(100),DELS(100),FPPW(100)
      COMMON/BL20/ RTHTR, UEIN,ROIN,GAMAT
      DIMENSION C(3)
      DATA C1,C2,C3,C4,C5,C6 /66.4663,-3.357287,12.31885,48447.19 ,
     1    -19886.08 , 1. /
C
C     - - - - - - - - - - - - - - - * - - - - - - - - - - - - - - -
C
      K=NX-1
      IIGR=0
      IF(LG16.NE.0) GO TO 50
      WRITE(6,9010)
      LSP=1
      GO TO 1800
   50 IF(V(1,2) .LE. 0.)  GO TO 600
      AG1=UE(NX)*RHOI/RMUI
      IF(FK .EQ. 1.) GO TO 55
      RX12=AG1*XS(NX)
      RTH12=AG1*THETA(NX)
      GO TO 57
   55 RTH12=AG1*THETA(NX)*ROL(NX)
      S2=0.
      DO 56 I=2,NX
      DELS3= XS(I)-XS(I-1)
   56 S2=S2+(ROL(I)**2)*(DELS3)
      RX12=AG1*S2
   57 CONTINUE
      IF(IGTR.NE.0) GO TO 60
      RX1=1.E-05 * RX12
      RTH1=RTH12
      IGTR=1
      GO TO 1800
   60 RX2=1.E-05 * RX12
      RTH2=RTH12
      IF(RX2 .LT. 1.) GO TO 550
      RTHE9 = C1 + C2*RX2 + C6*SQRT(C3*RX2*RX2+C4*RX2+C5)
      RTD = RTHE9-RTH2
      IF(RTD.GT.0. .AND. RTD.GE.10.)  GO TO 550
      IF(ABS(RTD) .GF. 10.)  GO TO 65
      IGTR=3
      ROT1=0.
      ROT2=RX2
      GO TO 95
C ----
   65 IGTR=3
      AG1=RTH2-((RTH2-RTH1)/(RX2-RX1))*RX2
      AG2=(RTH2-RTH1)/(RX2-RX1)
      AG3=AG2+3.357287
      AG4=AG1-66.4663
      C(1)=12.31885-AG3*AG3
      C(2)=48447.19-2.0*AG3*AG4
      C(3)=-19886.08-AG4*AG4
      BSM4AC = C(2)*C(2) - 4.*C(1)*C(3)
      IF(BSM4AC .GF. 0.)  GO TO 70
      WRITF(6,7000) NX
      GO TO 550
   70 ROT1 = (-C(2) + SQRT(BSM4AC))/(2.*C(1))
      ROT2 = (-C(2) - SQRT(BSM4AC))/(2.*C(1))
      IF(ROT1.LE.0.,AND.ROT2.LF.0.) GO TO 550
      IF(ROT1.GT.0.AND.ROT2.GT.0.) GO TO 80
      IF(ROT1.GT.0.) RX=1.F05*ROT1
      IF(ROT2.GT.0.) RX=1.F05*ROT2
      GO TO 100
   80 IIGR=1
      RX=ROT1 * 1.F05
      GO TO 100
   90 IIGR=2
      RX=ROT2 * 1.E05
C ----
      GO TO 100
   95 RX=1.F05*ROT2
      XTR=XS(NX)
      GO TO 200
  100 UETR = UE(NX) + (UE(NX)-UE(K))*(1.E-05*RX-RX1)/(RX2-RX1)
      XTR=RX*RMUI/(RHOI*UETR)
      IF(XTR-XS(NX)) 300,200,500
  200 WRITE(6,6010) NX
      GO TO 700
  300 IF(XTR .LE. XS(K))  GO TO 500
      WRITE(6,6020) XTR
      GO TO 1000
  500 IF(IIGR.EQ.1) GO TO 90
  550 IF(V(1,2).LE.0.) GO TO 600
      IF(IGTR.EQ.0) GO TO 1800
      RX1=RX2
      RTH1=RTH2
      CF0 = CF1
      CFSUM0 = CFSUM
      RETURN
  600 IGTR=2
      LG17 = 0
      IF(V(1,2).LT.0.) GO TO 800
      WRITF(6,6030) NX
  700 NXT=NX
      WRITF(6,6050) NXT
      CF1 = CF0
      CFSUM = CFSUMO
      IF(LG17 .EQ. 0)  GO TO 1800
      ROIN = 0.
      UEIN = 0.
      GO TO 1800
  800 XTR=XS(NX)-(XS(NX)-XS(K))* V(1,2)/(V(1,2)-V(1,1) )
      WRITE(6,6040) XTR
C ----
 1000 NXT=NX
      WRITE(6,6050) NXT
      CF1 = CF0
      CFSUM = CFSUMO
      IF(LG17 .EQ. 0)  GO TO 1800
      ROIN = XS(NX) - XTR
      UEIN = ROIN/UE(NX)
      IF(FK .NE. 0.)  ROIN = ROIN/RO(NX)
 1800 RETURN
C ----
 6010 FORMAT(1H1////////// 30X,
     1  34HTRANSITION HAS OCCURRED AT STATION, I3/)
 6020 FORMAT(1H1////////// 30X,
     1  30HTRANSITION HAS OCCURRFD AT S =, F12.6 /)
 6030 FORMAT(1H1///// 45X,38HLAMINAR SEPARATION OCCURRED AT STATION, I3,
     1 /)
 6040 FORMAT(1H1/////// 45X,34HLAMINAR SEPARATION OCCURRED AT S =,F12.6)
 6050 FORMAT(1H0,35X,33HTURBULENT FLOW STARTED WITH NTR = ,I3 ////)
 7000 FORMAT(1H1///40X,39HATTEMPT TO FIND X(TR) FAILED AT STATION ,I3/)
 9010 FORMAT(1H ,24H** ERROR IN TRFLAG INPUT, /)
 2000 CONTINUE                                                        TRNS  141C
      END                                                             TRNS  142
      SUBROUTINE SLOPE(NPX,XC, YC,DYDX, MER)
C
C **  SUBROUTINE SLOPE
C     COMPUTE THE DERIVATIVE DYDX FROM X VS Y INPUT
C
      DIMENSION  XC(150),YC(150),DYDX(150)
      DIMENSION X(300), Y(300),  XY(301)
      XY(1) = NPX
      IF(MER .NE. 0)  GO TO 20
      NP2M1 = NPX
      DO 10 I=1,NPX
      X(I) = XC(I)
      Y(I) = YC(I)
   10 CONTINUE
      GO TO 80
   20 NP2 = 2*NPX
      NP2M1 = NP2-1
      DO 40 I=1,NPX
      XY(I*2) = XC(I)
      XY(2*I+1) = YC(I)
   40 CONTINUE
      NLQ= 2
      DO 50 I=2,NP2,2
      X(I-1) = XY(I)
      Y(I-1) = XY(I+1)
      IF(I .EQ. NP2)  GO TO 50
      X(I) = (XY(I)+XY(I+2))*.5
      CALL INS1(X(I),XY , Y(I),NLQ,NER)
   50 CONTINUE
   80 DO 200 I=1,NP2M1
      IF(I .GT. 1) GO TO 100
      DYDX(I) = (Y(I+1)-Y(I)) / (X(I+1)-X(I))
      GO TO 200
  100 IF(I .LT. NP2M1)  GO TO 150
      DYDX(I) = (Y(I)-Y(I-1)) / (X(I)-X(I-1))
      GO TO 200                                                         SLOP  036
  150 IF(Y(I-1).EQ.Y(I) .AND. Y(I).EQ.Y(I+1))   GO TO 180               SLOP  037
      A1 = (X(I)-X(I+1)) / ((X(I-1)-X(I))*(X(I-1)-X(I+1)))              SLOP  038
      A2 = (2.*X(I)-X(I+1)-X(I-1)) / ((X(I)-X(I-1))*(X(I)-X(I+1)))      SLOP  039
      A3 = (X(I)-X(I-1)) / ((X(I+1)-X(I-1))*(X(I+1)-X(I)))              SLOP  040
      DYDX(I) = A1*Y(I-1) + A2*Y(I) + A3*Y(I+1)                         SLOP  041
      GO TO 200                                                         SLOP  042
  180 DYDX(I) = 0.                                                      SLOP  043
  200 CONTINUE                                                          SLOP  044
      IF(MER .EQ. 0)  RETURN                                            SLOP  045
      DO 300 I=1,NPX                                                    SLOP  046
  300 DYDX(I) = DYDX(2*I-1)                                             SLOP  047
      RETURN                                                            SLOP  048
      END                                                               SLOP  049
      SUBROUTINE OTPT
C
C  ** SUBROUTINE OTPT
C     OUTPUT THE RESULTS OF THE B. L. CALCULATIONS
C
      COMMON  NX,NP,NPPR,JI,IT,NRVP,LSP,NPM1,JI1,JIM1,NTC,NXT,NXW,NXM
     1      ,TITLE(15)
      COMMON LG16,LG17,LG18,LG32,LG40
     1      , IGOL, IGOT, IGOW, IGON, IGCV, IGEG, IGNP, IGRC, IGTR
      COMMON /HEADR/ CASE, IPAGE
      COMMON/BLC1/F(100,2),U(100,2),V(100,2),ETA(100),DELETA(100)
      COMMON/BLC3/XI(100),XS (100),ETAINF(100),BETA(100)
      COMMON/BLC5/EM(100,2),EDV(100,2),E(100,2),EB(100,2),VPRT(100)
      COMMON/BL12/TI,RMI,UI,RI,PR,PRT,FK,RL,RMUI,RHOI,PSI,HE
     1             ,UE(100),RO(100),TW(100),QW(100),RP(100),FW(100)
     2             ,BR(100),TE(100),RHOE(100),RMUE(100),GW(100),GPW(100)
     3             ,RF1(100),RF2(100),YS (100),IGX1(100),FPW(100),ROL(100)
      COMMON /BL14/ RX1,RTH1, CF0,CF1,CF2,CFSUM0,CFSUM,
     1       THETA(100),DELS(100),FPPW(100)
      COMMON /BL16/ NTYPE,IOUT
      COMMON /BL17/ RX(100),CFA(100),CF(100),ETAE(100),CDI(100),ST(100),
     1        INP(100)
      COMMON/BL19/C(100,2),G(100,2),GP(100,2),
     1            RHO(100),RMU(100),TVCT(100)
      COMMON/RADIUS/ ROMAX
      DIMENSION  Y(100)
C
C     - - - - - - - - - - - - - - - - * - - - - - - - - - - - - - -
C
      IF(IOUT .NE. 0) GO TO 900
      IPRT = (NP+18)/30
      IF(NP .LT. 30) IPRT=1
      A1 =0.
      UEUI = UE(NX)/UI
      THETA(NX) = 0.
      DELS(NX) = 0.
      RX(NX)=XS(NX)*RHOI*UE(NX)/RMUI
      RTHETA = 0.
      H=0.
      CF(NX) = 0.
      USTUF = 0.
      YPLUS = 0.
      UPLUS = 0.
      IF(NX .EQ. 1)  CF1 = 0.
      IF(NX .EQ. 1)  CFSUM = 0.
      CFA(NX) = 0.
      ST(NX) = 0.
      TNP(NX)=NP
      FTAE(NX) = ETA(NP)
      FPPW(NX) = V(1,2)
C----
      IF(XI(NX) .EQ. 0.) GO TO 300
      A1 = SQRT(2.*XI(NX))/( RHOI   *UF(NX)*ROL(NX))
      JINP = NP
      SUM1=0.0
      SUM2 = 0.
      F1=0.
      F3=1.0
      DO 90 J=2,JINP
      F2 = U(J,2)*(1.0-U(J,2))
      SUM1 = SUM1 + (F1+F2)/2.*DFLETA(J-1)
      F1=F2
   90 CONTINUE
      THETA(NX) = SUM1*A1
      DELS(NX) = A1*(ETAE(NX)    + F(1,2) - F(JINP,2))
      RTHETA = RX(NX)*THETA(NX)/XS(NX)
      H = DELS(NX)/THETA(NX)
      CF(NX) = SQRT(2./XI(NX))* RMUI   *V(1,2)*ROL(NX)        *F(1,2)
      CF(NX) = ABS(CF(NX))
      USTUF = SQRT(CF(NX)/2.)
      IF(FK .GE. 1.0) GO TO 200                                         OTPT  071
      IF(NX .EQ. 1)  CF1 = CF(NX)*UEUI*UEUT                             OTPT  072
      IF(NX .FQ. 1)  GO TO 300                                          OTPT  073
      CF2 = CF(NX) *UEUI*UEUT                                           OTPT  074
      CFSUM   = CFSUM  +  (CF1+CF2)*(YS(NX)-YS(NX-1))*.5                OTPT  075
      IF(XI(NX-1) .EQ. 0.)  CFSUM = 2.*THETA(NX)                        OTPT  076
      CFA(NX) = CFSUM / (YS(NX)-YS(1))                                  OTPT  077
      CF1 = CF2                                                         OTPT  078
      GO TO 300                                                         OTPT  079
  200 CONTINUE                                                          OTPT  080
      IF(NX .EQ. 1) CF1=CF(NX)*UEUI*UEUI*RO(NX)                         OTPT  081
      IF(NX .EQ. 1) GO TO 300                                           OTPT  082
      CF2=CF(NX)*UEUI*UEUI*RO(NX)                                       OTPT  083
      CFSUM=CFSUM + (CF1+CF2) * (YS(NX)-YS(NX-1))*.5*RL                 OTPT  084
      IF(XI(NX-1) .EQ. 0.) CFSUM=2.*THETA(NX)                           OTPT  085
      CFA(NX)=CFSUM * 2. /(ROMAX*ROMAX)                                 OTPT  086
      CF1=CF2                                                           OTPT  087
  300 IF(LG32.EQ.1 .AND. IGX1(NX).EQ.0)  GO TO 420                      OTPT  088
      LCMAX=42                                                          OTPT  089
      LC=60                                                             OTPT  090
      SUMY = 0.                                                         OTPT  091
      F1=1.0                                                            OTPT  092
      DO 400 J=1,NP                                                     OTPT  093
      IF(LC .LT. LCMAX) GO TO 340                                       OTPT  094
      LC=1                                                              OTPT  095
      CALL HEAD                                                         OTPT  096
      WRITF(6,2000)NX,YS(NX)                                            OTPT  097
      IF((LG32.EQ.0.OR.IGX1(NX).NE.0))                   WRITE(6,2100)  OTPT  098
      IF(LG32 .EQ. 2)  WRITE(6,2150)                                    OTPT  099
  340 IF(J .EQ. 1)  GO TO 350                                           OTPT  100
      F2=1./TVCT(J)                                                     OTPT  101
      SUMY = SUMY + (F1+F2)/2.*DFLETA(J-1)                              OTPT  102
      F1 = F2                                                           OTPT  103
  350 I = 1 + ((J-1)/IPRT)*IPRT                                         OTPT  104
      IF(J.NE.I .AND. J.NE.NP)  GO TO 400                               OTPT  105
      Y(J) = SUMY*A1
      IF(USTUE .EQ. 0.)  GO TO 370
      YPLUS = Y(J)*UE(NX)*RHOE(NX)*USTUE/RMUE(NX)
      UPLUS = U(J,2)/USTUE
      LC=LC+1
  370 WRITE(6,6060)
     1 J,ETA(J),F(J,2),U(J,2),V(J,2),Y(J),YPLUS,UPLUS,FDV(J,2)
  400 CONTINUE
  420 CONTINUE
C----
C ----
  700 IF(LG32.EQ.1 .AND. IGX1(NX).EQ.0)  GO TO 800
      WRITE(6,3000)
      WRITE(6,3200) NX,XS(NX), THETA(NX), DELS(NX),CF(NX),V(1,2),GW(NX)
      WRITE(6,3250) YS(NX), RX(NX),RTHETA, H, CFA(NX),GPW(NX),ST(NX)
  800 IF(IOUT .EQ. 0) GO TO 1800
C----
  900 CALL HEAD
      WRITE(6,3500) TITLE
      LCMAX=45
      LC=1
      WRITE(6,4000)
      IF(LSP .EQ. 1 .AND. NX .GT. 1) NX=NX-1
      DO 1000 I=1,NX
      IF(LC .LT. LCMAX) GO TO 940
      CALL HEAD
      WRITE(6,4000)
      LC=1
  940 RTHETA=0.
      H=0.
      IF(XI(I) .EQ. 0.)  GO TO 950
      RTHETA = RX(I)*THETA(I)/XS(I)
      H = DELS(I)/THETA(I)
  950 WRITE(6,4200) I,XS(I),THETA(I),DELS(I),CF(I),FPPW(I),GW(I),INP(I)
      WRITE(6,4250) YS(I), RX(I),RTHETA, H, CFA(I),GPW(I),ST(I),ETAE(I)
      LC=LC+3
 1000 CONTINUE
      DO 1003 I=1,NX
 1003 DELS(I)=DELS(I)/RL
      WRITE(2) NX
      WRITE(2) (DELS(K), K=1,NX)
C     THE CALCULATION OF THE BASE DRAG IS DONE AT THIS POINT USING
C     A METHOD GIVEN IN HOERNERS= BOOK ON AERODYNAMIC DRAG
      IF(RO(NX) .LT. ROMAX) GO TO 1010
      CDBASE=.029/SQRT(CFA(NX))
      GO TO 1020
 1010 CFAB=CFA(NX)
      CDBASE=.029*(RO(NX)/ROMAX)**3/SQRT(CFAB)
 1020 WRITE(6,1030) CDBASE
 1030 FORMAT(1H0, 77H THE BASE DRAG FOR THIS CONFIGURATION BASED ON THE
     1MAXIMUM FRONTAL AREA IS = ,  F15.8)
 1800 RETURN
C - - - -
 2000 FORMAT(1H ,3X,11HSTATION NO.,I3,30X, 5HS/C =,F12.6 /)
 2100 FORMAT(1H0,2X,1HI,7X,3HETA,11X,1HF,14X,2HFP,14X,3HFPP,14X,1HY,14X,
     1       5HYPLUS,12X,5HUPLUS,12X, 4HEPS+ /)
 2150 FORMAT(1H0,2X,1HI,7X,3HETA,11X,1HF,14X,2HFP,14X,3HFPP,11X,
     1        7HY/THETA,9X,5HYPLUS,12X,5HUPLUS,12X,4HEPS+, /)
 2200 FORMAT(1H0,2X,1HI,7X,3HETA,11X,1HF,14X,2HFP,14X,3HFPP,14X,1HY,12X,
     1       4HW/WE,12X,5H  WP ,12X,4HEPS+, /)
 2300 FORMAT(1H0,2X,1HI,7X,3HETA,11X,1HG,14X,2HGP,14X,2H Y,13X,4H   T ,
     1       14X,4H PRT,12X,2HMU,14X,2HM ,/)
 2400 FORMAT(1H0,2X,1HI,7X,3HETA,11X,1HG,14X,2HGP,11X,7HY/THETA, 10X,
     1       4HT/TE,11X,10H   PRT    ,8X, 6HMU/MUE,10X, 4HM/ME, / )
 3000 FORMAT(1H0// 7X, 1HN, 12X,4H  S ,13X,5HTHETA,12X,4HDELS,14X,2HCF,
     1      14X,4HFPPW,14X,2HGW, /1H ,5X, 3HX/C,12X,2HRX,13X,6HRTHETA,14X
     2      ,1HH,15X,3HCFA,14X,3HGPW, 14X, 2HST, / )
 3200 FORMAT(1H /1H ,I7,4X, 6E17.6 )
 3250 FORMAT(1H ,F11.5,6E17.6 )
 3500 FORMAT(1H ,42X,14HOUTPUT SUMMARY,15A4/ )
```

```
4000 FORMAT(1HO/ 7X,1HN,12X,4H  S ,13X,5HTHETA,12X,4HDELS,14X,2HCF,14X,  OTPT  176
    1      4HFPPW,14X,2HGW,15X,4HIMAX,/1H ,5X,3HX/C,12X,2HRX,13X,        OTPT  177
    2      6HRTHETA,14X,1HH,15X,3HCFA,14X,3HGPW,14X,2HST,14X,6HETAINF,/)  OTPT  178
4200 FORMAT(1H /1H ,I7,4X, 6E17.6, 8X, I4)                               OTPT  179
4250 FORMAT(1H ,F11.6,6E17.6,  4X, F11.6)                                OTPT  180
6060 FORMAT(1H ,I3,2X,F10.6,  7E16.6      )                              OTPT  181
9000 CONTINUE                                                            OTPT  182C
     END                                                                 OTPT  183
```

```
      SURROUTINE HFAD
C
C **  SUBROUTINE HFAD
C
      COMMON /HEADR/ CASF, IPAGE
      WRITE (6,100)  CASE
      IPAGE = IPAGE + 1
      RETURN
  100 FORMAT(1H1,/1H ,2X,6H CASE ,A4,21X,51H****** CEBECI-KELLER BOUNDAR
     1Y LAYER PROGRAM *******, 23X , 13HPROGRAM K99A )
      END
      OVFRLAY(AXSY,6,0)                                                 SMOT  001C
      PROGRAM SMOOTH                                                    SMOT  002C
C     SUBROUTINE SMOOTH                                                 SMOT  003I
C     ****************************************************************** SMOT  004
C     THIS SUBROUTINE CONTROLS THE SMOOTHING OF THE INPUT COORDINATES   SMOT  005
C     ****************************************************************** SMOT  006
      DIMENSION X(100),Y(100),XO(100),YO(100)                           SMOT  007
      REWIND 1                                                          SMOT  008
      REWIND 10                                                         SMOT  009
      READ(5,1) NPTS ,ITAPE                                             SMOT  010
      IF(ITAPE .NE. 0) GO TO 5                                          SMOT  011
      READ(5,2) (X(I),I=1,NPTS)                                         SMOT  012
      READ(5,2) (Y(I),I=1,NPTS)                                         SMOT  013
      GO TO 7                                                           SMOT  014
    5 READ(1) (X(I),I=1,NPTS)                                           SMOT  015
      READ(1) (Y(I),I=1,NPTS)                                           SMOT  016
    7 CONTINUE                                                          SMOT  017
      CALL SM5PT(X,Y,XO,YO,NPTS)                                        SMOT  018
      WRITE(10,10)     (XO(I),I=1,NPTS)                                 SMOT  019
      WRITE(10,10)     (YO(I),I=1,NPTS)                                 SMOT  020
      REWIND 10                                                         SMOT  021
    1 FORMAT(2I4)                                                       SMOT  022
    2 FORMAT(6F10.0)                                                    SMOT  023
   10 FORMAT(6F10.6)                                                    SMOT  024
C     RETURN                                                            SMOT  025I
   20 CONTINUE                                                          SMOT  026C
      END                                                               SMOT  027
      SUBROUTINE SM5PT (XI,YI,XO,YO,N)
C
C  THIS ROUTINF USFS THE OPTIMUM 5 POINT SMOOTHING METHOD.  A 3 POINT
C  METHOD TS USED AT THE END POINTS.
C
      DIMENSION XI(1),YI(1),XO(1),YO(1)
C
C
      J = 2
      I = -1
   10 XO(J+I) = XI(J+I)
      YO(J+I) = YI(J+I)
      XO(J) = 0.25*(XI(J-1) + 2.0*XI(J) + XI(J+1))
      YO(J) = 0.25*(YI(J-1) + 2.0*YI(J) + YI(J+1))
      IF (I .EQ. 1) GO TO 20
      J = N-1
      I = 1
      GO TO 10
C
  20  CONTINUE
      N2 = N - 2
      DO 30 J=3,N2
      XO(J) = (-XI(J-2)+4.0*XI(J-1)+10.0*XI(J)+4.0*XI(J+1)-XI(J+2))*
     1   0.0625
  30  YO(J) = (-YI(J-2)+4.0*YI(J-1)+10.0*YI(J)+4.0*YI(J+1)-YI(J+2))*
     1   0.0625
C
      RETURN
      END
      OVERLAY(AXSY,7,0)                                                 ITRT  001C
      PROGRAM ITERAT                                                    ITRT  002C
C     SUBROUTINE ITERAT                                                 ITRT  003I
      DIMENSION S(100),DELLS(100),X(100),Y(100),SURF(100),SINAL(100),   ITRT  004
     1 COSAL(100),SMD(100),DFLSTR(201),TSINAL(201),TCOSAL(201),         ITRT  005
     2 XNEW(100),YNEW(100) ,DESU(100)                                   ITRT  006
      REWIND 15                                                         ITRT  007
      REWIND 2                                                          ITRT  008
C     SURFACE DISTANCES FROM BOUNDARY LAYER INPUT                       ITRT  009
      READ(2) N                                                         ITRT  010
      READ(2) (S(I),I=1,N)                                              ITRT  011
C     BOUNDARY LAYER DISPLACEMENT THICKNESS                             ITRT  012
      READ(2) NN                                                        ITRT  013
      READ(2) (DELLS(I),I=1,NN)                                         ITRT  014
C     THIS DO LOOP IS USED IN CASE THE BOUNDARY LAYER DOES NOT          ITRT  015
C     CALCULATE A COMPLETE BOUNDARY DISPLACEMENT THICKNESS ARRAY        ITRT  016
C     DUE TO TURBULENT BOUNDARY LAYER SEPARATION                        ITRT  017
      DO 10 I=NN,N                                                      ITRT  018
   10 DELLS(I) = DELLS(NN)                                              ITRT  019
C     THE COORDINATES OF THE TRANSFORMED BODY ARE READ IN AT THIS POINT ITRT  020
      READ(15) NTS                                                      ITRT  021
      READ(15) (X(I),I=1,NTS)                                           ITRT  022
      READ(15) (Y(I),I=1,NTS)                                           ITRT  023
C     THE SURFACE DISTANCE OF THE INPUT BODY COORDINATES ARE CALCULATED ITRT  024
      SURF(1)=0.0                                                       ITRT  025
      DO 30 I=2,NTS                                                     ITRT  026
      DESURF=SQRT((X(I)-X(I-1))**2+(Y(I)-Y(I-1))**2)                    ITRT  027
   30 SURF(I)=DESURF+SURF(I-1)                                          ITRT  028
      S(N)=SURF(NTS)                                                    ITRT  029
C     NOTE S(I) IS DURFACE DISTANCE FROM LEADING EDGF TO TRAILING EDGE  ITRT  030
C     BASED ON COORDINATES ASSOCIATED WITH THE BOUNDARY LAYER SOLUTION  ITRT  031
C     SURF(I) IS SURFACE DISTANCE FROM LEADING EDGE TO TRAILING EDGE    ITRT  032
C     BASED ON THE X AND Y COORDINATES OF THE TRANSFORMED ORIGINAL      ITRT  033
C     BODY AT WHICH THE VALUE OF DISPAACEMENT THICKNESS IS TO BE ADDED  ITRT  034
C     NEXT THE COSINE AND SIN OF THE LOCAL SURFACE ANGLES               ITRT  035
```

```
C     ARE FOUND AT THE MIDPOINTS OF THE X AND Y COORDINATES
      NTSM=NTS-1
      NTSMM=NTS+1
      DO 40 I=1,NTSM
      DESU(I)=SQRT((X(I+1)-X(I))**2+(Y(I+1)-Y(I))**2)
      SINAL(I+1) = (Y(I+1)-Y(I))/DESU(I)
   40 COSAL(I+1) = (X(I+1)-X(I))/DESU(I)
      SINAL(1) = SINAL(2)
      COSAL(1) = COSAL(2)
      SINAL(NTSMM)=SINAL(NTS)
      COSAL(NTSMM)=COSAL(NTS)
C     THE SURFACE DISTANCE CORRESPONDING TO EACH OF THESE SIN AND COSINE
C     PAIRS ARE CALCULATED HERE
      SMD(1)=0.0
      SMD(2)=.5*SURF(2)
      DO 50 I=2,NTSM
   50 SMD(I+1) = SMD(I)+.5*(SURF(I+1)-SURF(I-1))
      SMD(NTSMM)=SMD(NTS)+.5*(SURF(NTS)-SURF(NTSM))
C     DISPLACEMENT THICKNESS AT THE VALUES OF S AND Y ARE FOUND HERE
      CALL TABLE1(N,S,DELLS,DELSTB)
      CALL TABLE1(NTSMM,SMD,SINAL,TSINAL)
      CALL TABLE1(NTSMM,SMD,COSAL,TCOSAL)
C     NEW COORDINATES ARE FORMED HERE
      DO 60 I=1,NTS
      SURFF=SURF(I)
      CALL INS1(SURFF,DELSTB,DELST,1,NER)
      CALL INS1(SURFF,TSINAL,SINU,1,NER)
      CALL INS1(SURFF,TCOSAL,COSU,1,NER)
      XNEW(I)=X(I)-DELST*SINU
   60 YNEW(I)=Y(I)+DELST*COSU
      XNEW(1) = X(1) - DELLS(2)
C     IF SEPARATION HAS OCCURRED, THE BODY IS MODIFIED TO ACCOUNT
C     FOR THIS BY ADDING A CIRCULAR RADIUS TO CREATE A SEPARATION BUBBLE
      IF(NN .EQ. N) GO TO 80
      XNEW1=XNEW(NN-3)
      XNFW2=XNEW(NN-2)
      XNFW3=XNFW(NN-1)
      YNFW1=YNFW(NN-3)
      YNFW2=YNEW(NN-2)
      YNEW3=YNEW(NN-1)
      CALL CIRCLE(XNFW1,XNFW2,XNEW3,YNFW1,YNEW2,YNFW3,RADIUS,XCENT,
     1  YCENT,DYDX)
      NNN=NN-1
      DO 70 I=NNN,NTS
      DELTA=X(I)-XCENT
      IF(DFLTA .GE. 0.) KM=I
      IF(DELTA .GE. 0.)  GO TO 90
      YCIR=(RADIUS **2 - DFLTA **2)
      IF(DYDX .GE. 0.) GO TO 65
      YNFW(I)=YCENT-SQRT(YCIR)
      GO TO 70
   65 YNFW(I)=YCENT+SQRT(YCIR)
   70 CONTINUE
      GO TO 80
   90 CONTINUE
      DO 100 I=KM,NTS
  100 YNEW(I)=YNEW(KM-1)
   80 CONTINUE
C     IF THE BODY RADIUS BECOMES EQUAL TO THE DISPLACEMENT THICKNESS AT
C     SOME POINT THEN THE NEW BODY IS MODIFIFD TO KEEP THE SAME AREA
C     DUE TO THE DISPLACEMENT THICKNESS
      NXSL=NTS/2
      DO 105 K=NXSL,NTS
      DYNEW=YNEW(K)-Y(K)
      IF(DYNEW .GE. Y(K)) KSL=K-2
      IF(DYNFW .GE. Y(K)) GO TO 110
  105 CONTINUE
      GO TO 115
  110 DAREA=3.14159*(YNEW(KSL)**2-Y(KSL)**2)
      DO 112 I=KSL,NTS
      ARFA=3.14159*Y(I)*Y(I)+DARFA                                      ITRT  106
  112 YNFW(I)=SQRT(AREA/3.14159)                                        ITRT  107
  115 WRTTF(6,4)                                                        ITRT  108
      WRTTF(6,5) (XNEW(I),YNEW(I),I=1,NTS)                              ITRT  109
C     XNFW AND YNEW ARE THE VISCOUS COORDINATES WHICH SHOULD BE WRITTEN ITRT  110
C     ON TAPE AND TRANSFERRED TO THE SMOOTHING ROUTINE                  ITRT  111
      REWIND 1                                                          ITRT  112
      WRTTE(1) (XNFW(I),I=1,NTS)                                        ITRT  113
      WRTTE(1) (YNEW(I),I=1,NTS)                                        ITRT  114
    4 FORMAT(1H ,10X,4HXNEW,20X,4HYNFW)                                 ITRT  115
    5 FORMAT(2F20.8)                                                    ITRT  116
      RETURN                                                            ITRT  117
      END                                                               ITRT  118
      SUBROUTINE TABLE1(N,X,Y,TABLE)
      DIMENSION X(100),Y(100),TABLE(201)
C     THIS ROUTINE SETS UP TABLES FOR INPUT TO SUBROUTINE INS1
C     N IS THE NUMBER OF VALUES OF X AND Y TO BE PUT INTO ARRAYS
      J=2
      DO 200 I=1,N
      TABLE(J)=X(I)
      TABLE(J+1) = Y(I)
      J=J+2
  200 CONTINUE
      TABLE(1)=N
      RETURN
      END
      SUBROUTINE CIRCLE (X1,X2,X3,Y1,Y2,Y3,R,XCENT,YCENT,DYDX)
      XY1=-(X1*X1+Y1*Y1 )
      XY2=-(X2*X2+Y2*Y2 )
      XY3=-(X3*X3+Y3*Y3 )
      E1=(XY1-XY2)*(X2-X3)-(XY2-XY3)*(X1-X2)
      F2=(Y1-Y2)*(X2-X3)-(Y2-Y3)*(X1-X2)
      E=F1/E2
      CENTK=-E/2.
      D=((XY2-XY3)-E*(Y2-Y3))/(X2-X3)
      H=-D/2.
      F=XY1-D*X1-E*Y1
      R=SQRT(H*H+CENTK**2-F)
      DYDX=-(X3-H)/(Y3-CENTK)
      C=1./R
      DYDXS=DYDX*DYDX
      CC=C*(1.+DYDXS)**1.5
      CC=ABS(CC)
      DDYDX=ABS(DYDX)
      PHI=ATAN(DDYDX)
      IF(DYDX .GE. 0.) GO TO 20
      XCFNT=X3+R*SIN(PHI)
      YCENT=Y3+R*COS(PHI)
      GO TO 30
   20 XCENT=X3+R*SIN(PHI)
      YCENT=Y3-R*COS(PHI)
   30 CONTINUE
      RETURN
      END
```

REFERENCES

1. Thwaites, B.: Incompressible Aerodynamics. Oxford at the Clarendon Press, p. 62, 1960.

2. Callaghan, J. G. and Beatty, T. D.: A Theoretical Method for the Analysis and Design of Multi-element Airfoils. Journal of Aircraft Volume 9, No. 12, December 1972.

3. Stevens, W. A.; Goradia, S. H.; and Braden, J. A.: Mathematical Model for Two-Dimensional Multi-Component Airfoils in Viscous Flow, NASA CR-1843, July 1971.

4. Bhateley, I. C. and McWhirter, J. W.: Development of Theoretical Method for Two-Dimensional Multi-element Airfoil Analysis and Design. Air Force Flight Dynamics Laboratory, TR-72-96, Part I, August 1972.

5. Hess, J. L.: Calculation of Potential Flow about Arbitrary Three-Dimensional Lifting Bodies. McDonnell Douglas Report No. MDC J5679-01, done under Contract No. N00019-71-C-0524 for Naval Air Systems Command.

6. Hess, J. L. and Smith, A. M. O.: Calculation of Potential Flow about Arbitrary Bodies. Progress in Aeronautical Sciences, Volume 8, Pergamon Press, New York, 1966.

7. Smith, A. M. O. and Pierce, J.: Exact Solution of the Neumann Problem. Calculation of Plane and Axially Symmetric Flows about or within Arbitrary Boundaries. Douglas Aircraft Company Report No. 26988, April 1958. (A brief summary is contained in the preceedings of the third U.S. National Congress of Applied Mechanics, Brown University, 1958).

8. Cebeci, T. and Smith, A. M. O.: A Finite-Difference Solution of the Incompressible Turbulent Boundary Layer Equations by an Eddy-Viscosity Concept. Computation of Turbulent Boundary Layers, AFOSR-IFP Stanford Conference, Volume 1, Stanford University Press, Stanford, California, 1968, pp. 346-356.

9. Faulkner, S.; Hess, J. L.; and Giesing, J. P.: Comparison of Experimental Pressure Distributions with those Calculated by the Neumann Program. Douglas Aircraft Company Report LB 31831, December 1964.

10. Bauer, A. B.; Smith, A. M. O.; and Hess, J. L.: Potential Flow and Boundary Layer Theory as Design Tools in Aerodynamics. Canadian Aeronautics and Space Journal, Volume 16, No. 2, February 1970.

11. Hess, J. L.: Numerical Solution of the Integral Equation for the Neumann Problem with Application to Aircraft and Ships. Presented to Symposium on Numerical Solution of Integral Equations with Physical Applications. SIAM Meeting, October 11, 12, 13, 1971, University of Wisconsin, Madison, Wisconsin, Douglas Engineering Paper 5987.

12. Hess, J. L.: Calculation of Potential Flow about Bodies of Revolution Having Axes Perpendicular to the Freestream Direction. Douglas Aircraft Company Report No. ES 29812, October 1, 1960.

13. Hess, J. L.: Extension of the Douglas-Neumann Program for Axisymmetric Bodies to Include Calculation of Potential, Non-uniform Cross Flow, Added Mass, and Conductor Problems. Douglas Aircraft Company Report No. 31765, September 1, 1964. This work performed for Naval Ordnance Test Station, Pasadena Annex under Contract No N60530-10053.

14. Hess, J. L.: Extension of the Douglas Axisymmetric Potential Flow Program to Include the Effects of Ring Vorticity with Application to the Program of Specified Tangential Velocity. Douglas Aircraft Company Report No. 33195, June 10, 1966. This work performed for Naval Ordinance Test Station, Pasadena Annex under Contract No. N60530-11208.

15. Hess, J. L.: Improved Ring-Source Formulas for Small Values of Distance from the Symmetry Axis. A Modification of the Douglas-Neumann Program for Axisymmetric Bodies. Douglas Aircraft Company Report No. 70002, August 1969.

16. Hess, J. L.: Modification of the Douglas-Neumann Program for Axisymmetric Bodies to Include the Direchlet Problem for a Potential that Varies as the Cosine of Twice the Circumferential Angle. Douglas Aircraft Company Report No. 70001, August 1969. This work performed under Air Force Contract AFO(694)-953 on the Reentry Systems Environment Protection Program.

17. Hess, J. L. and Schoor, C.: Extension of the Douglas-Neumann Axisymmetric Potential Flow Program to the Problem of a Ring Wing Having a Known Ring Vortex Wake Issuing from its Trailing Edge. McDonnell Douglas Report No. MDC J0741/01, April 25, 1970. This work performed for Naval Undersea Research and Development Center under Contract No. N66001-70-C-0436.

18. Probstein, R. F. and Elliott, D.: The Transverse Curvature Effect in Compressible Axially Symmetric Laminar-Boundary-Layer Flow. Journal of Aeronautical Sciences, March 1956.

19. Hartree, D. R. and Womersley, J. R.: A Method for the Numerical or Mechanical Solution of Certain Types of Partial Differential Equations. Procedure Royal Society Series A, Volume 161, No. 906, p. 353, August 1937.

20. Keller, H. B.: A New Difference Scheme for Parabolic Problems in "Numerical Solution of Partial Differential Equations." Volume II, Academic Press, New York, 1970.

21. Keller, H. B.; and Cebeci, T.: Simple Accurate Numerical Methods for Boundary Layers. I. Two-Dimensional Laminar Flows. Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics. Lecture Notes in Physics, Volume 8, Springer-Verlag, New York, 1971.

22. Keller, H. B.; and Cebeci, T.: Simple Accurate Numerical Methods for Boundary Layers. II. Two-Dimensional Turbulent Flows, AIAA Journal Volume 19, No. 9, pp. 1197-1200, September 1972.

23. Van Driest, E. R.: On Turbulent Flow Near a Wall. Journal Aeronautical Sciences, Volume 23, no. 11, p. 1007, November 1956.

24. Cebeci, T.: Eddy-Viscosity Distribution in Thick Axisymmetric Turbulent Boundary Layers. Journal of Fluids Engineering, June 1973, pp. 319 to 326.

25. Cebeci, T.: Kinematic Eddy Viscosity at Low Reynolds Numbers. AIAA Journal, Volume 11, No. 1, January 1973, pp. 102-104.

26. Coles, D.: The Turbulent Boundary Layer in a Compressible Fluid. Report R-403-PR., September 1962, Rand Corporation, Santa Monica, California.

27. Cebeci, T.: Laminar and Turbulent Incompressible Boundary Layers on Slender Bodies of Revolution in Axial Flow. Journal of Basic Engineering, Trans. ASME, Series D, Volume 92, No. 3, September 1979, pp. 545-554.

28. Cebeci, T.: Wall Curvature and Transition Effects in Turbulent Boundary Layers. AIAA Journal, Volume 9, No. 9, September 1971, pp. 1868-1870.

29. Chen, K. K. and Thyson, W. A.: Extension of Emmons' Spot Theory to Flows on Blunt Bodies. AIAA Journal, Volume 9, No. 5, pp. 821-825.

30. Emmons, H. W.: The Laminar-Turbulent Transition in a Boundary Layer. Journal of the Aerospace Sciences, Volume 18, Part I, 1950, p. 490.

31. Michel, R.: Etude de la Transition Sur Les Profils D'Aile-Establissement d'un critere de Determination du Point de Transition et Calcul de la Tratnee de Profil en Incompressible. Onera Rapport 1/1578A, July 1951.

32. Smith, A. M. O.: Transition, Pressure Gradient, and Stability Theory. Proceedings of the 9th International Congress of Applied Mechanics, Volume 4, 1956, p. 234.

33. Kaups, K.: Transition Prediction on Bodies of Revolution. McDonnell Douglas Report No. MDC J6530, prepared under U.S. Navy Undersea Center Contract No. N66001-74-C-0020, April 1974.

34. Cebeci, T.; Mosinskis, G. J.; and Smith, A. M. O.: Calculation of Viscous Drag and Turbulent Boundary Layer Separation on Two-Dimensional and Axisymmetric Bodies in Incompressible Flow. McDonnell Douglas Report No. MDC J0973-01, prepared under Contract No. N00014-70-C-0099, for the Naval Ship and Systems Command, November 1970.

35. Hoerner, S. F.: Base Drag and Thick Trailing Edges. Journal of the Aeronautical Sciences, 1959, p. 622.

36. Bauer, A. B.: Body of Revolution Drag Measurement and Results. McDonnell Douglas Corporation Report No. MDC J5167/01, December 1970.

37. Tomotike, S. and Imai, I.: On the Transition from Laminar to Turbulent Flow in the Boundary Layer of a Sphere. Aeronautical Research Institute of Tokyo University Report No. 167, p. 389-423, August 1938.

38. Achenbach, Elmar: Experiments on the Flow Past Spheres at Very High Reynolds Numbers. Journal of Fluid Mechanics Volume 54, Part 3, 1972, pp. 565-575.

39. Jacob, K.: Theoretische Berechung von Druckverteilung und Kraftbeiwerten Für Beliebige Profile bej Inkompressibler Strömung mit Ablösung. Ava-Bericht 67 A 62 (1967).

40. Beatty, T.D.: Prediction of Flows with and without Partial Separation. Masters Thesis presented to Mechanical Engineering Department, California State University, Long Beach, July 1972.

41. Hahn, M.; Rubbert, P. E.; and Mahal, A.: Evaluation of Separation Criteria and Their Application to Separated Flow Analysis. Air Force Flight Dynamics Laboratory Report No. AFFDL-TR-72-145, January 1973.

42. Cebeci, T.; Mosinskis, G. J.: and Kaups, K.: A General Method for Calculating Three-Dimensional Incompressible Laminar and Turbulent Boundary Layers. 1. Swept Infinite Cylinders and Small Cross Flow McDonnell Douglas Aircraft Corporation Report No. MDC J5694, prepared under Contract No. N00014-72-C-0111, for the Naval Ship Systems Command.

43. Chang, P.K.: Separation of Flow. Pergamon Press, 1970.

44. Wang, K. C.: Separation Patterns of Boundary Layer Over an Inclined Body of Revolution. AIAA Journal, Volume 10, No. 8, August 1972.

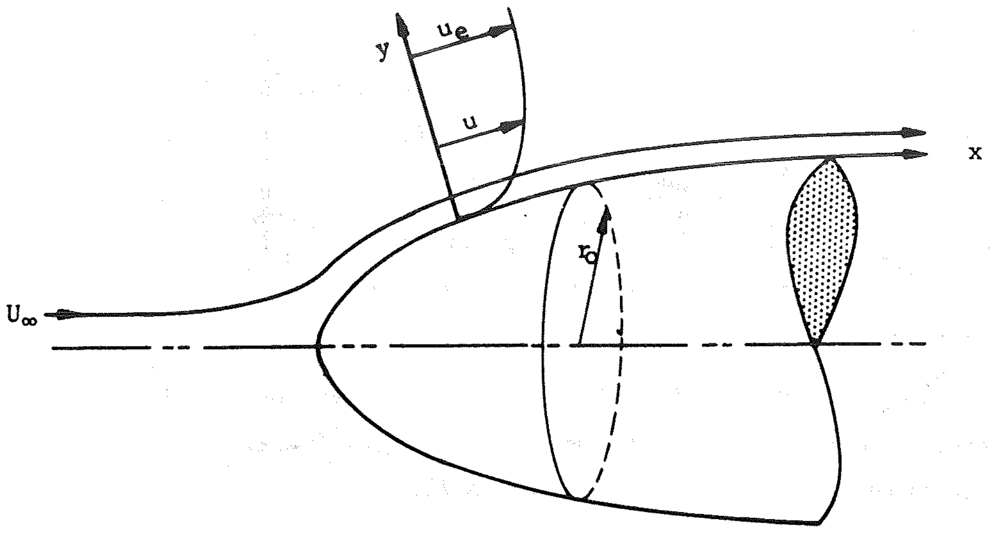


FIGURE 1. COORDINATE SYSTEM FOR THE BOUNDARY LAYER ON A BODY OF REVOLUTION

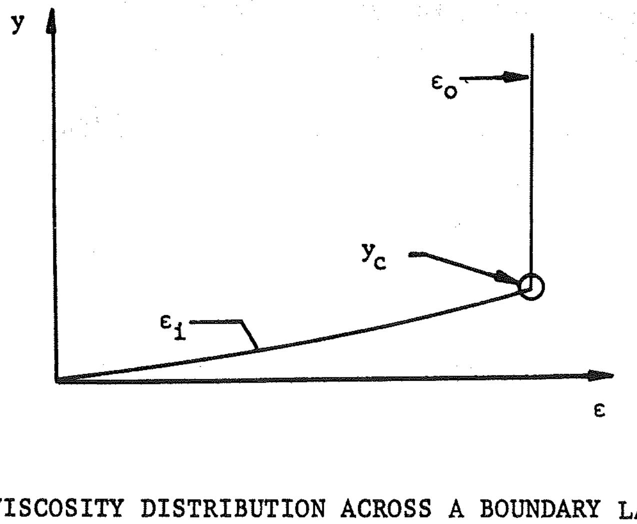


FIGURE 2. EDDY-VISCOSITY DISTRIBUTION ACROSS A BOUNDARY LAYER

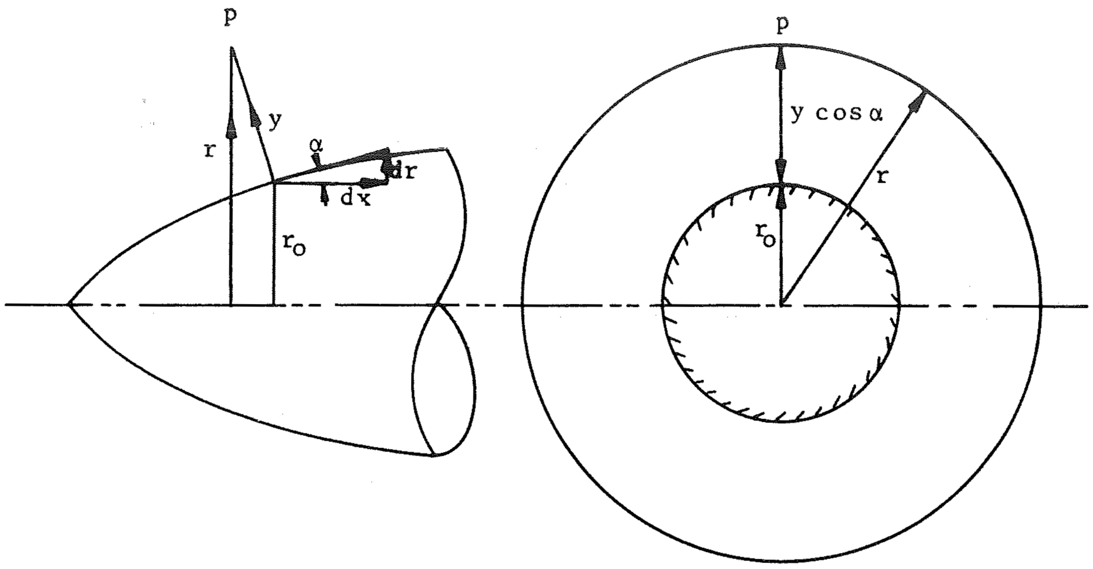


FIGURE 3. COORDINATES FOR AXIALLY SYMMETRIC BODY WITH THICK BOUNDARY LAYER

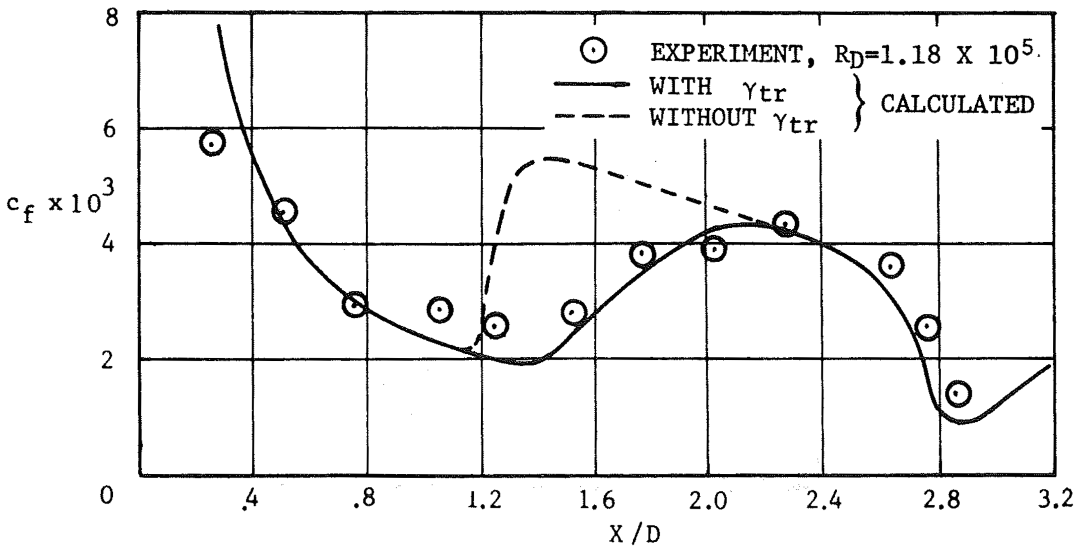


FIGURE 4. EFFECT OF TRANSITION REGION MODIFICATION ON THE SKIN FRICTION

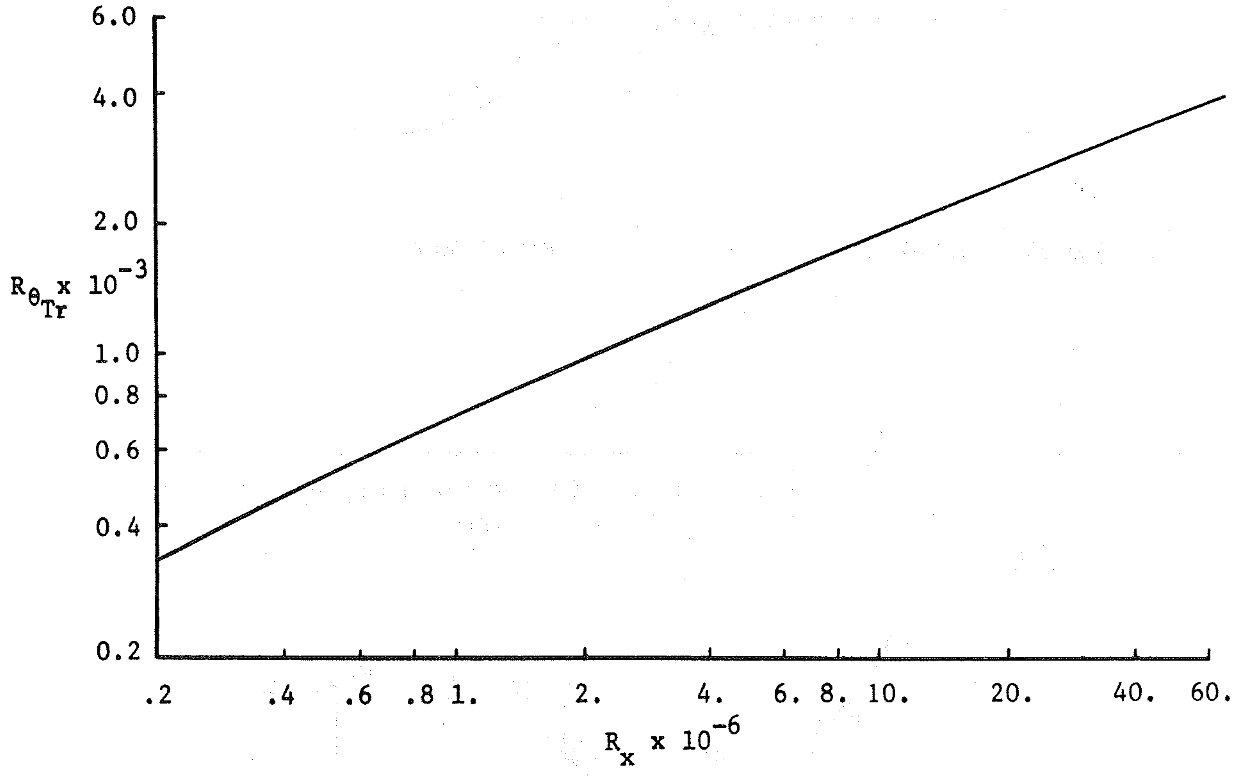


FIGURE 5. TRANSITION CORRELATION CURVE FROM REFERENCE 32

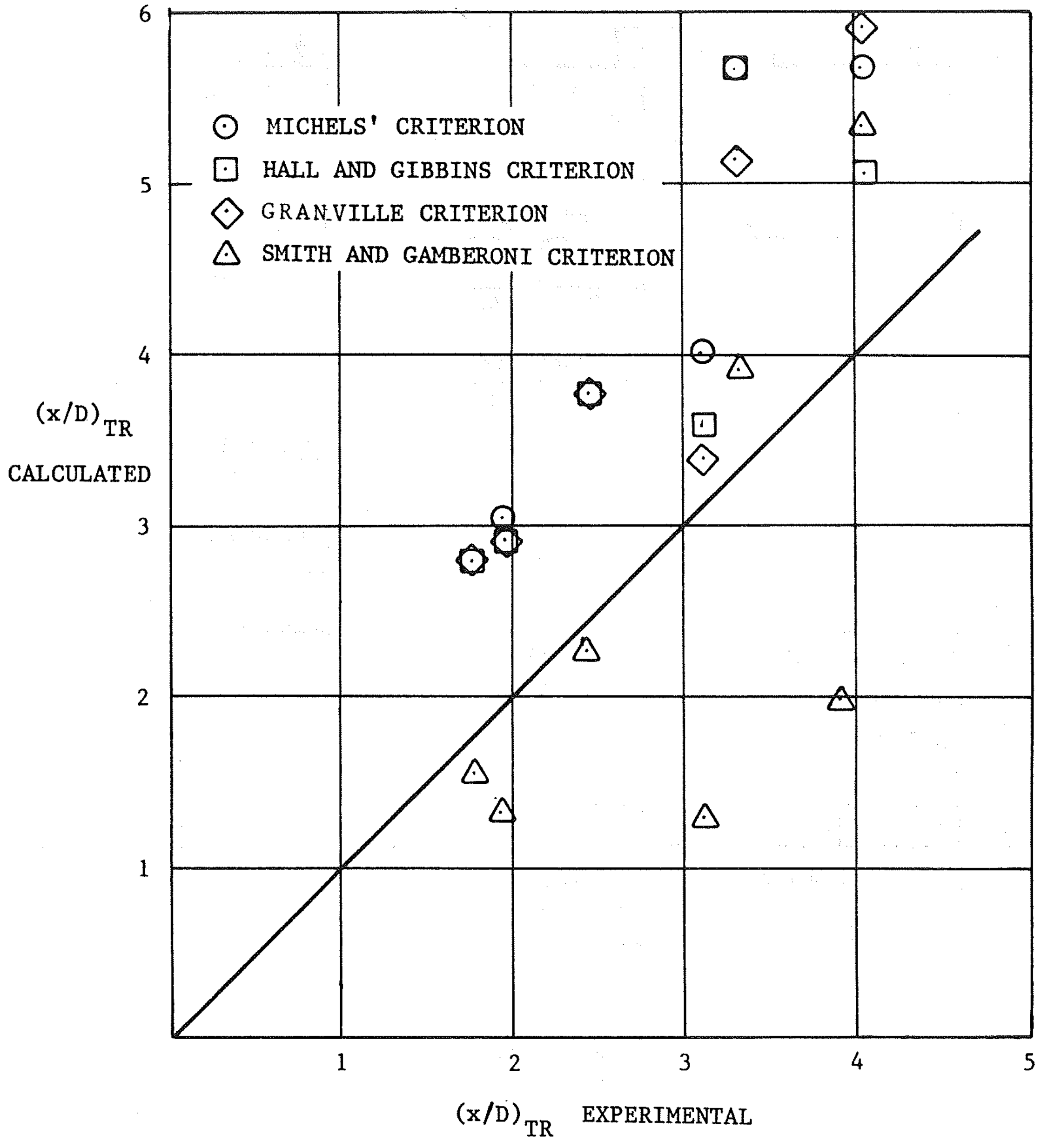


FIGURE 6. COMPARISON OF EXPERIMENTAL AND CALCULATED TRANSITION LOCATIONS FROM VARIOUS METHODS FOR FAVORABLE GRADIENT FLOWS

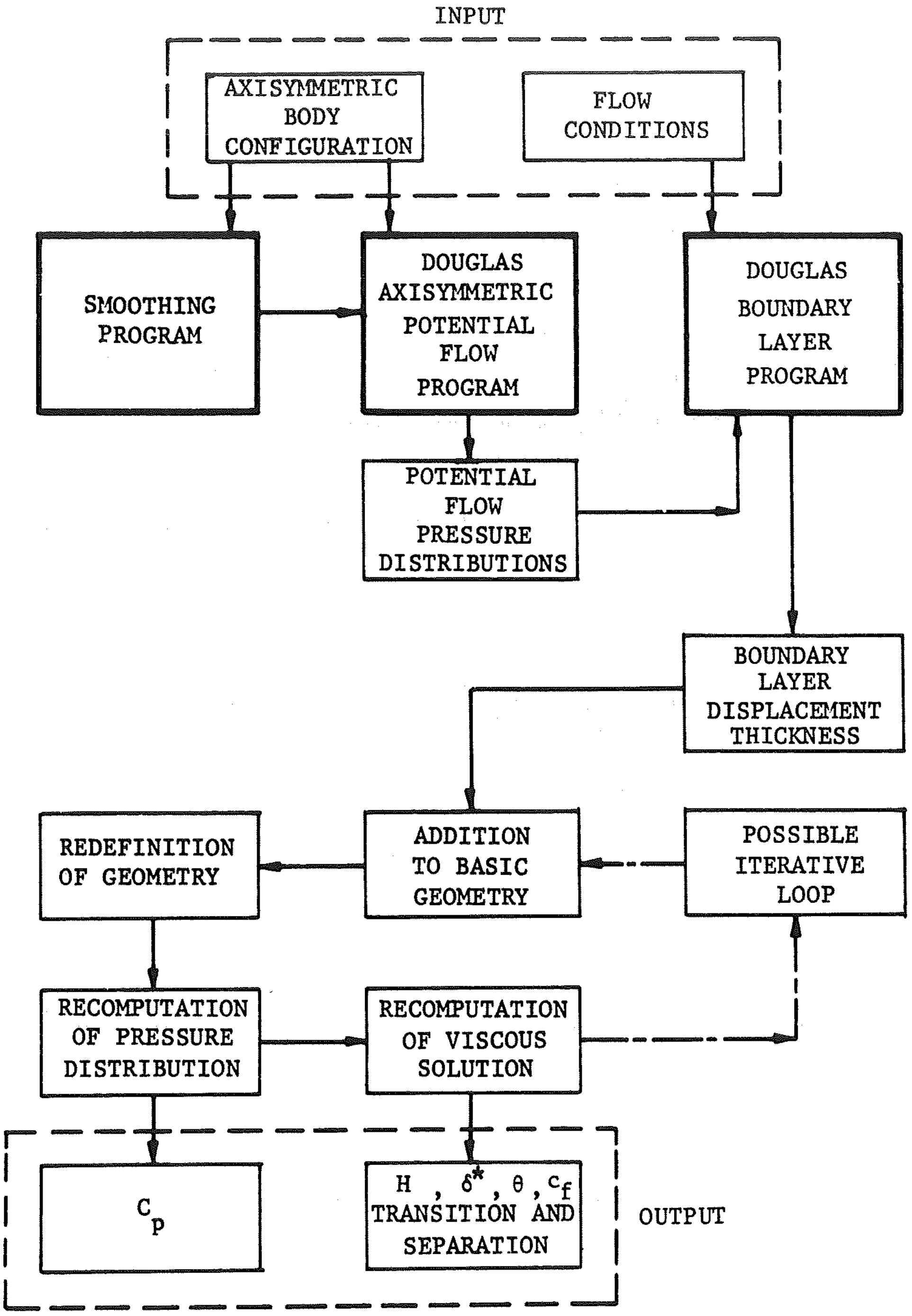

INPUT
AXISYMMETRIC BODY CONFIGURATION
FLOW CONDITIONS
SMOOTHING PROGRAM
DOUGLAS AXISYMMETRIC POTENTIAL FLOW PROGRAM
DOUGLAS BOUNDARY LAYER PROGRAM
POTENTIAL FLOW PRESSURE DISTRIBUTIONS
BOUNDARY LAYER DISPLACEMENT THICKNESS
REDEFINITION OF GEOMETRY
ADDITION TO BASIC GEOMETRY
POSSIBLE ITERATIVE LOOP
RECOMPUTATION OF PRESSURE DISTRIBUTION
RECOMPUTATION OF VISCOUS SOLUTION
C_p
H , δ^* , θ , c_f
TRANSITION AND SEPARATION
OUTPUT

FIGURE 7. FLOW DIAGRAM OF COMPUTER PROGRAM FOR AXISYMMETRIC ANALYSIS AND DESIGN METHOD (ADAM)

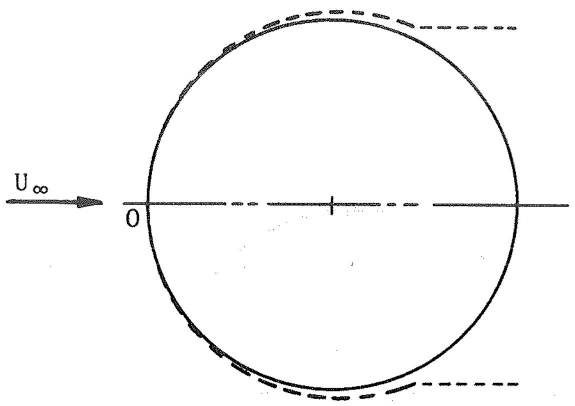


FIGURE 8. SCHEMATIC DIAGRAM OF CYLINDRICAL WAKE SHAPE USED TO MODEL SEPARATION

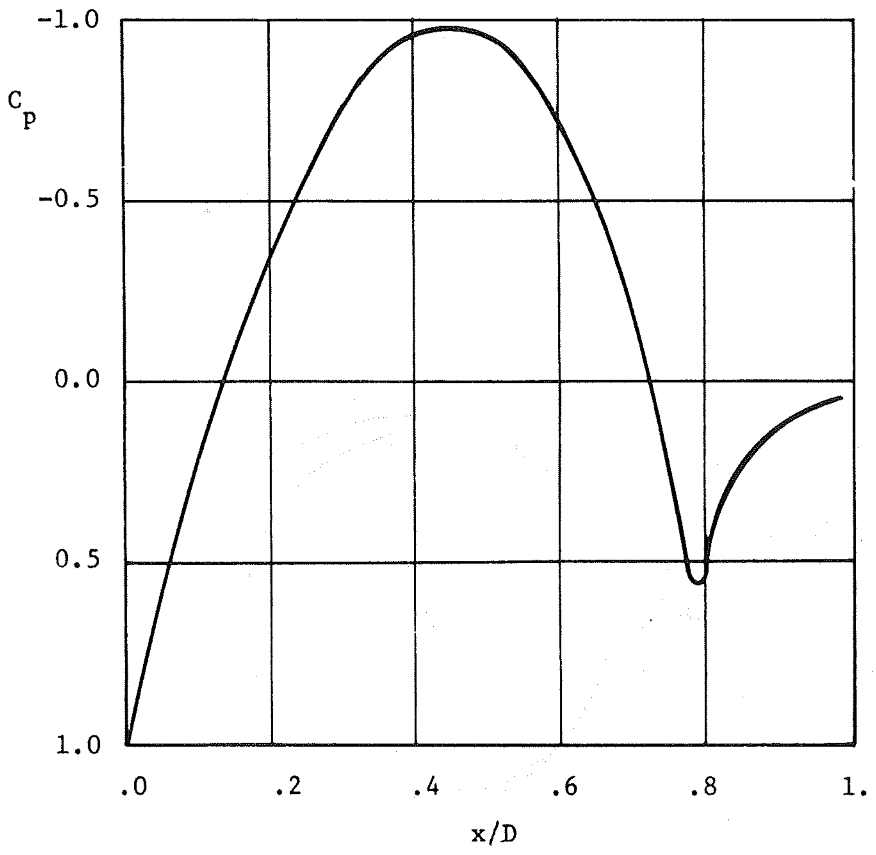


FIGURE 9. PRESSURE DISTRIBUTION FOR SPHERE IN SUPERCRITICAL REGION USING CYLINDRICAL SEPARATION MODEL

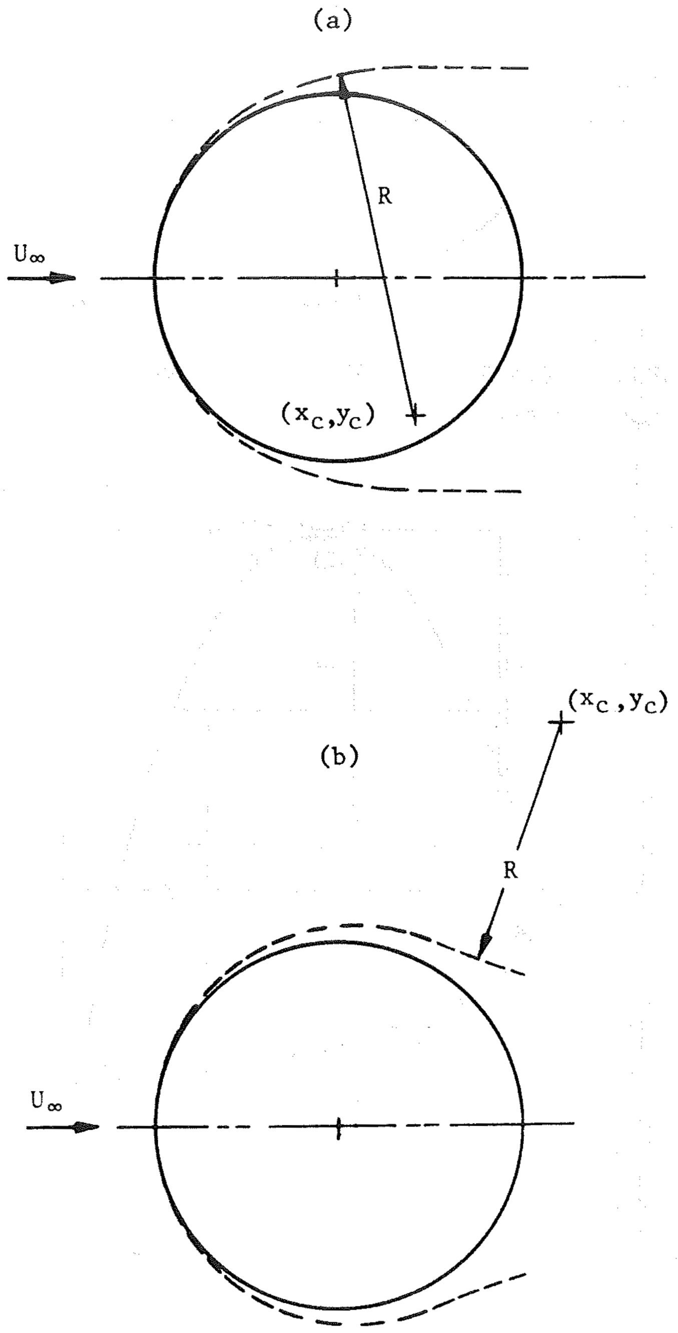


FIGURE 10. SCHEMATIC DIAGRAMS OF CIRCULAR ARC FAIRING USED IN THE MODEL FOR SEPARATED FLOW.

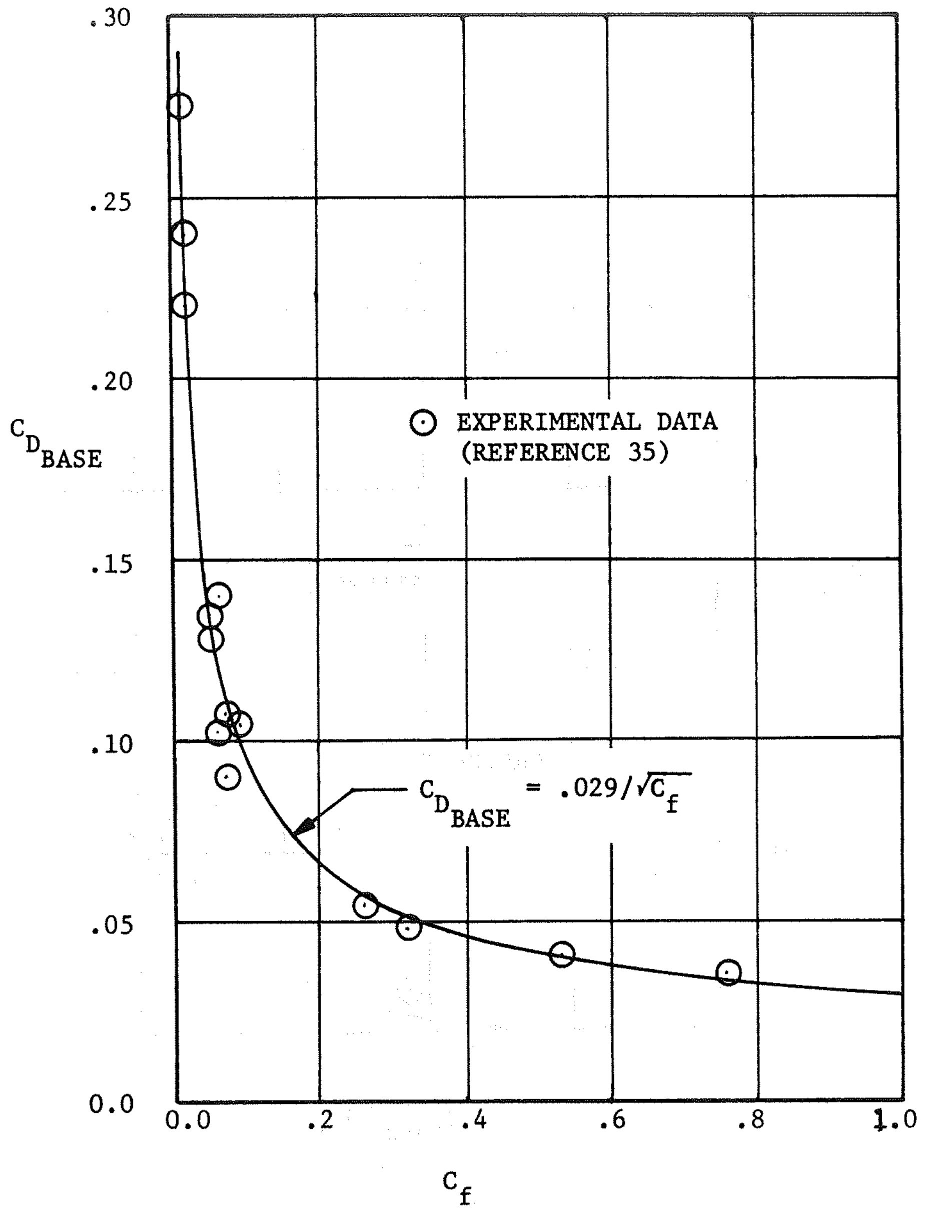


FIGURE 11 AXISYMMETRIC BASE DRAG AS A FUNCTION OF FOREBODY SKIN FRICTION COEFFICIENT

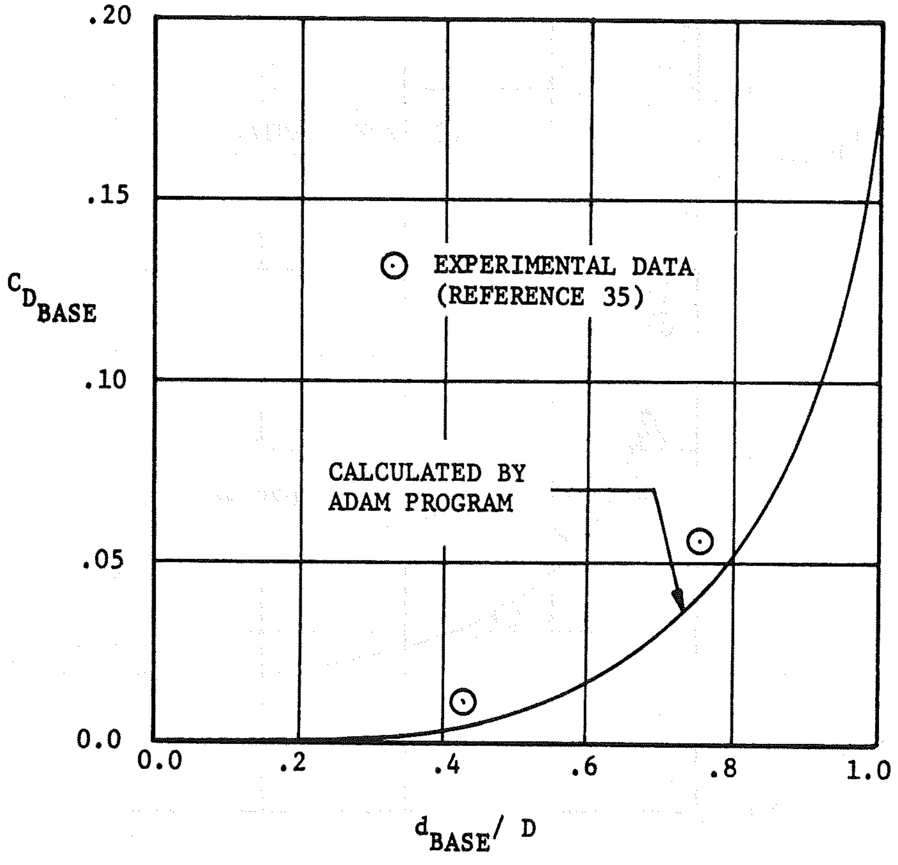


FIGURE 12 COMPARISON OF BASE DRAG CALCULATED BY ADAM TO EXPERIMENTAL DATA

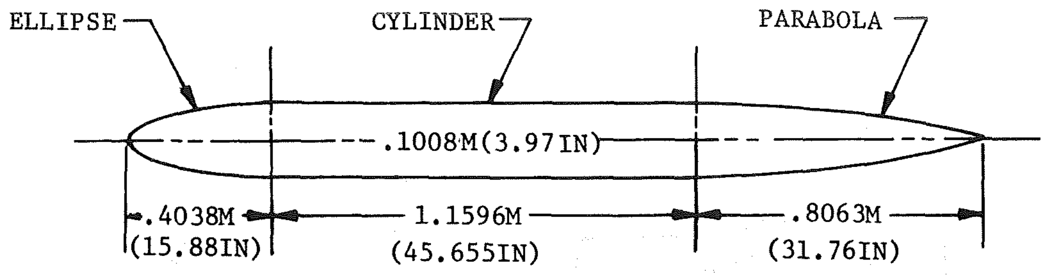


FIGURE 13. SCHEMATIC OF HIGH FINENESS RATIO BODY FROM REFERENCE 35

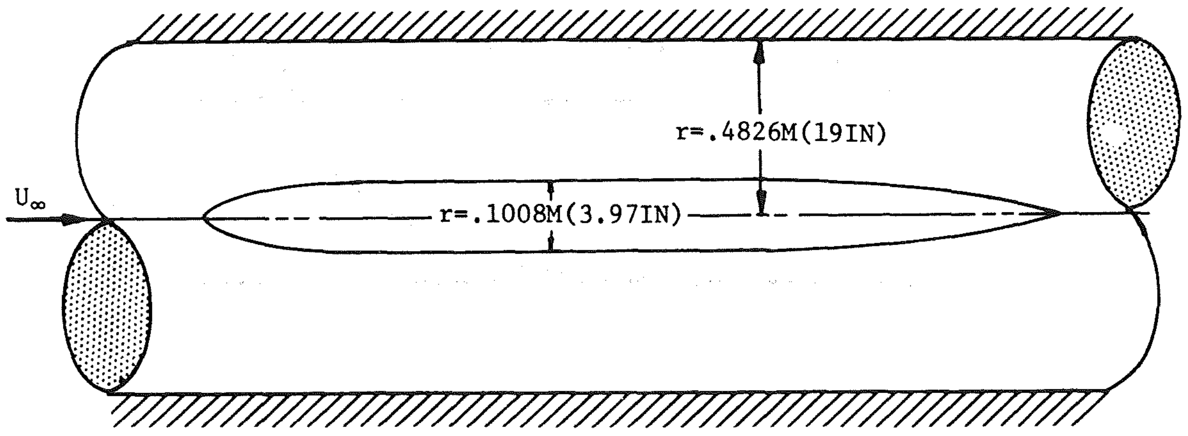


FIGURE 14. HIGH FINENESS RATIO BODY AND SIMULATED TUNNEL USED IN POTENTIAL FLOW PROGRAM TO ACCOUNT FOR WALL EFFECTS ON PRESSURE DISTRIBUTION

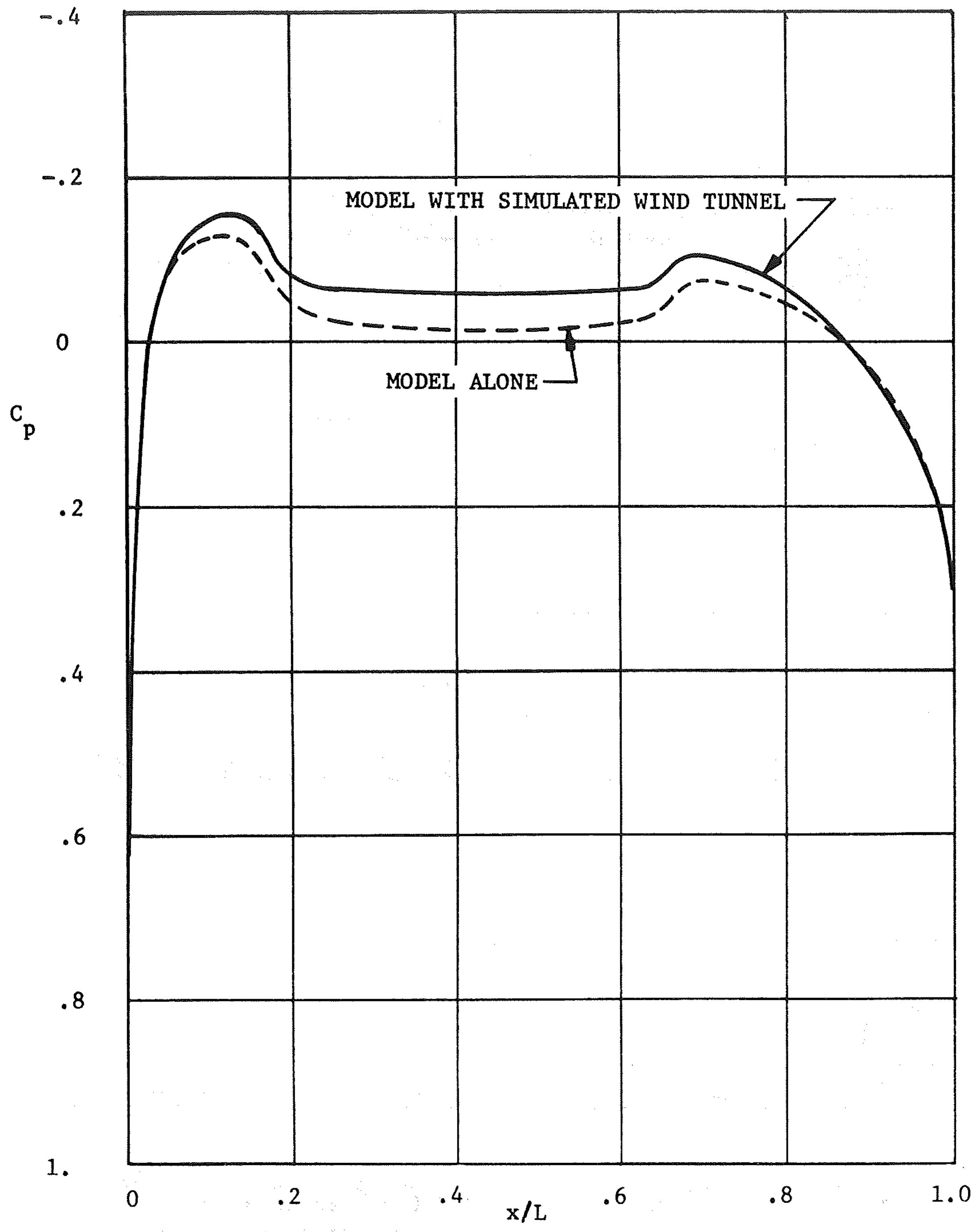


FIGURE 13. EFFECT OF WIND TUNNEL WALLS ON INVISCID PRESSURE DISTRIBUTION FOR HIGH FINENESS RATIO BODY AS CALCULATED BY POTENTIAL FLOW PROGRAM

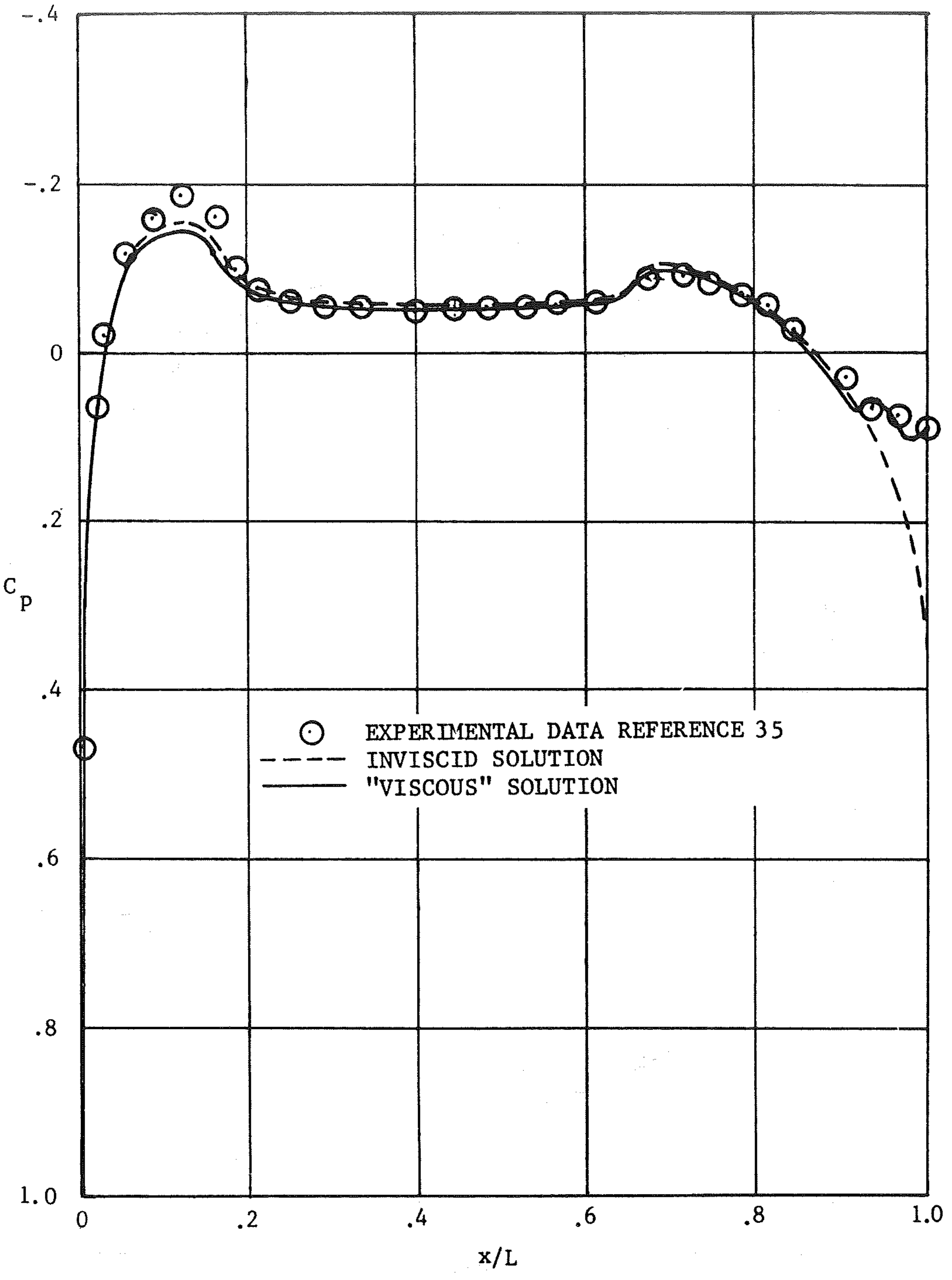


FIGURE 14. COMPARISON OF CALCULATED "VISCOUS" PRESSURE DISTRIBUTION FOR HIGH FINENESS RATIO BODY TO EXPERIMENTAL DATA

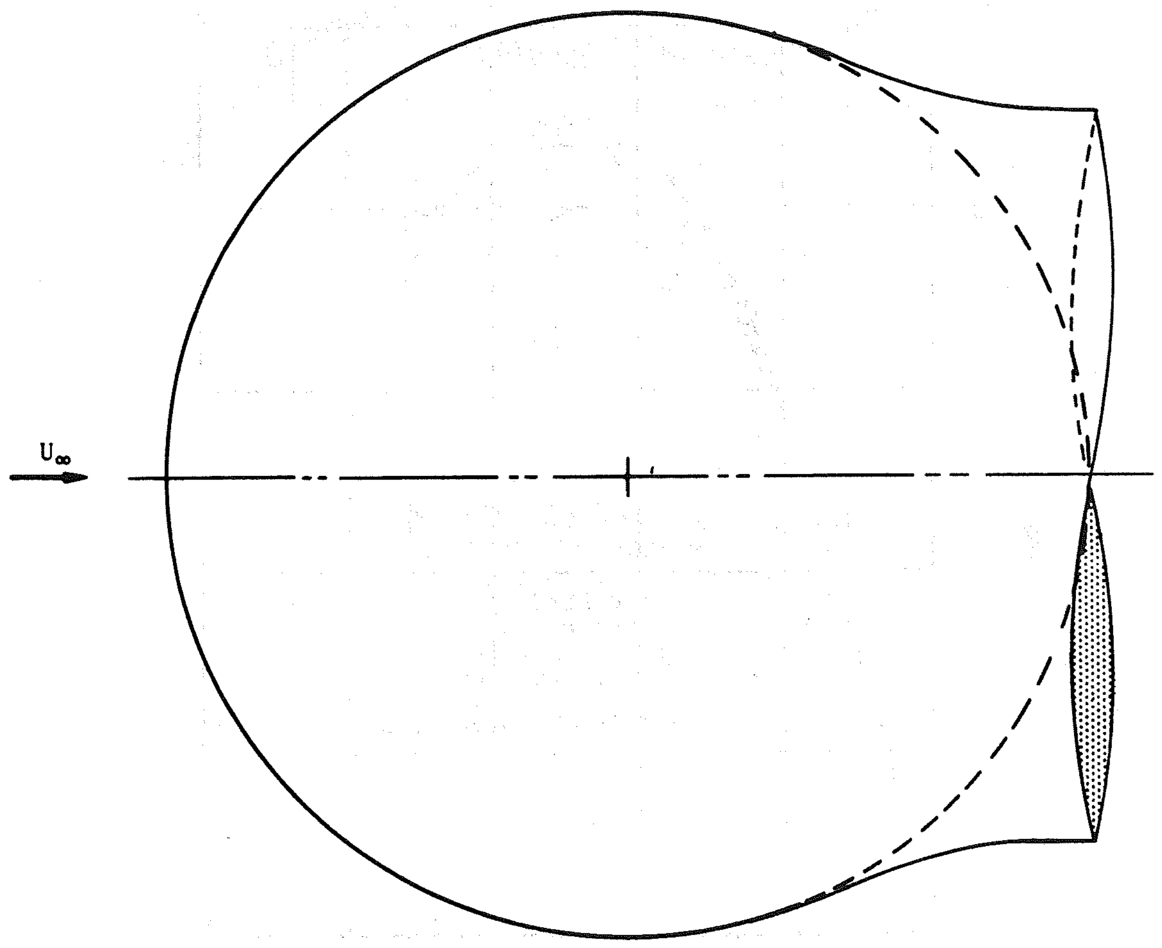


FIGURE 15. EQUIVALENT BODY INCLUDING SEPARATED WAKE USED TO CALCULATE "VISCOUS" FLOW ABOUT SPHERE IN SUPERCRITICAL REGIME

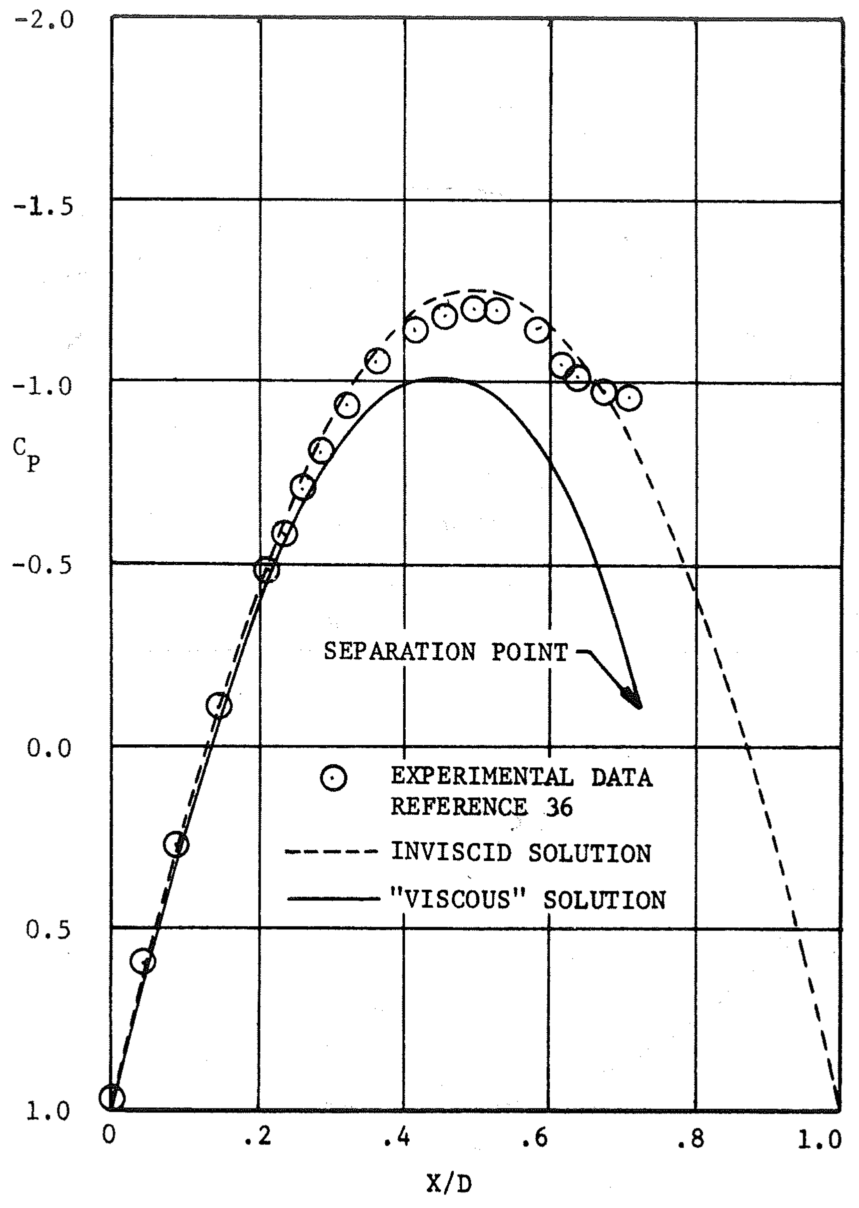


FIGURE 16. COMPARISON OF CALCULATED "VISCOUS" PRESSURE DISTRIBUTION FOR SPHERE IN SUPERCRITICAL REGIME TO EXPERIMENTAL DATA

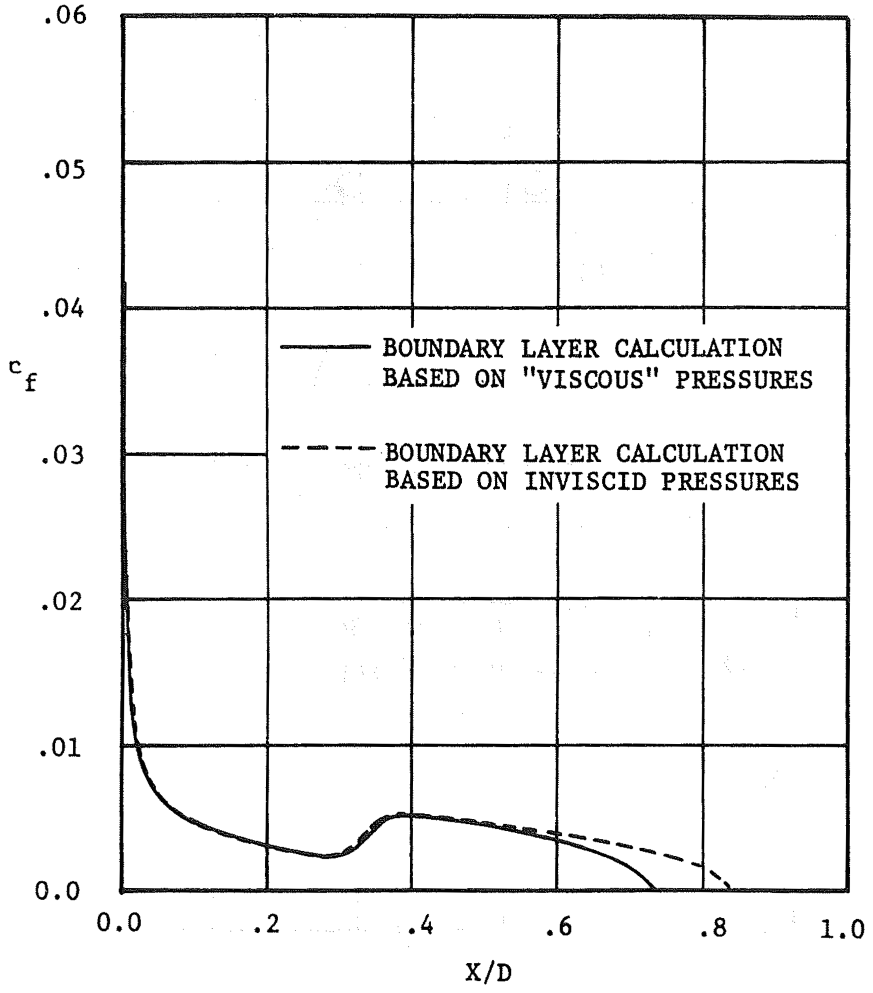


FIGURE . EFFECT OF "VISCOUS" MODELING ON CALCULATION OF LOCAL SKIN FRICTION COEFFICIENT FOR SPHERE IN SUPERCRITICAL REGIME

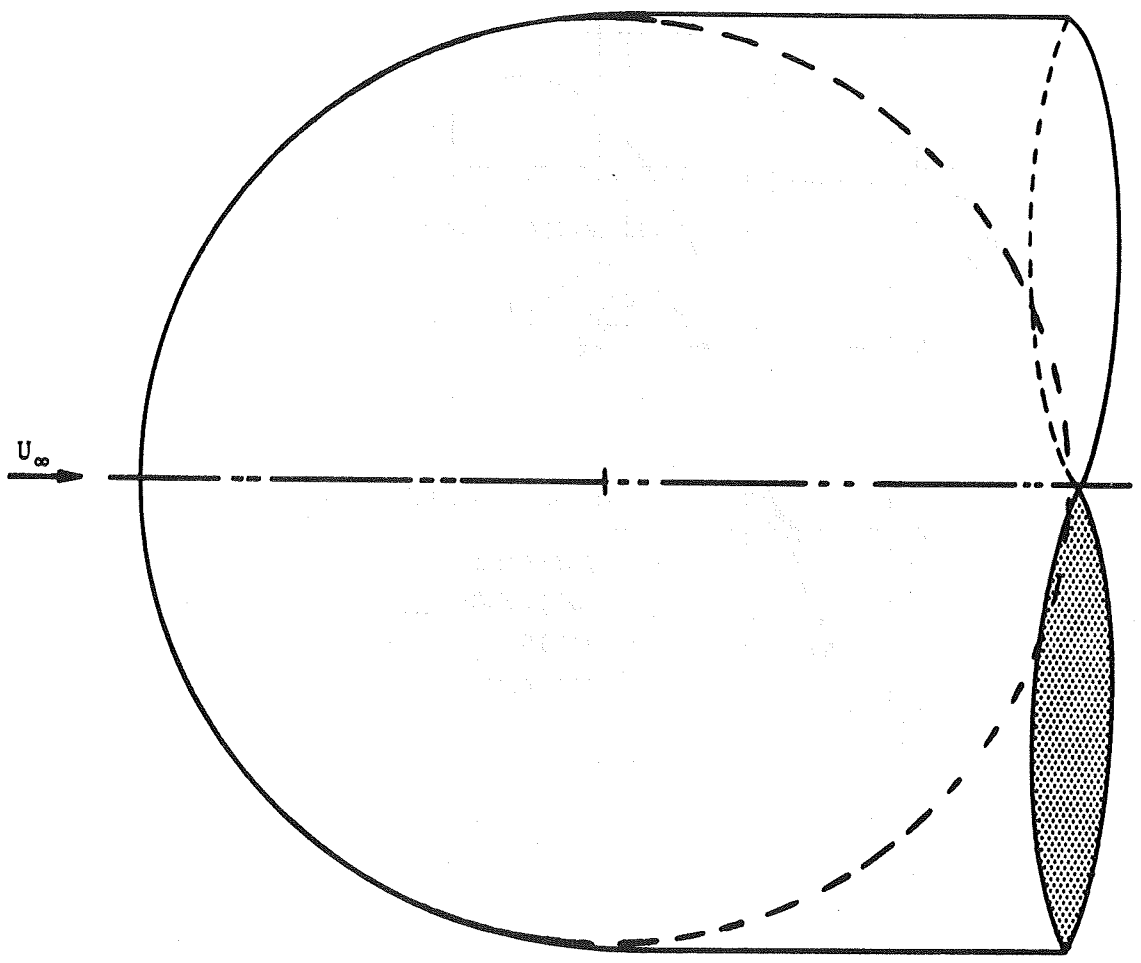


FIGURE 17. EQUIVALENT BODY INCLUDING SEPARATED WAKE USED TO CALCULATE "VISCOUS" FLOW ABOUT SPHERE IN SUBCRITICAL REGIME

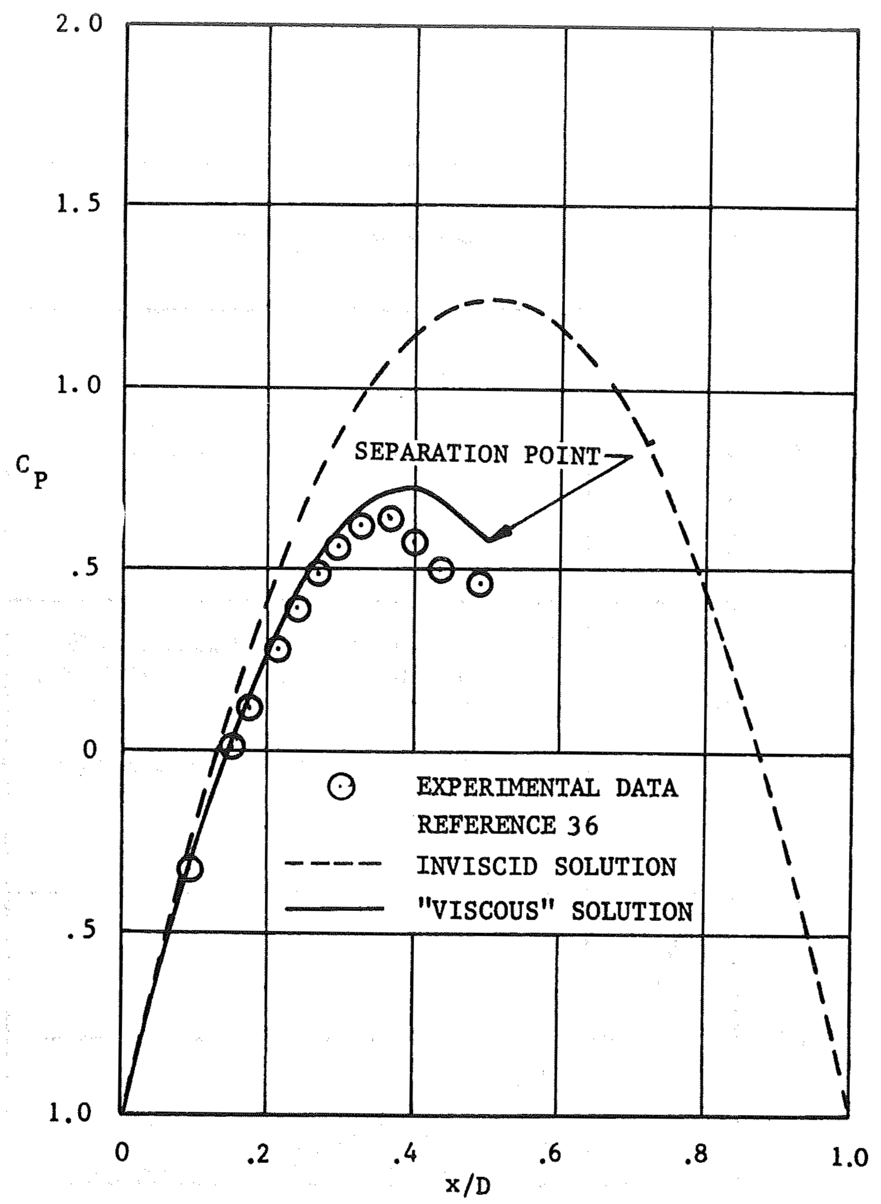


FIGURE 18. COMPARISON OF CALCULATED "VISCOUS" PRESSURE DISTRIBUTION FOR SPHERE IN SUBCRITICAL REGIME TO EXPERIMENTAL DATA

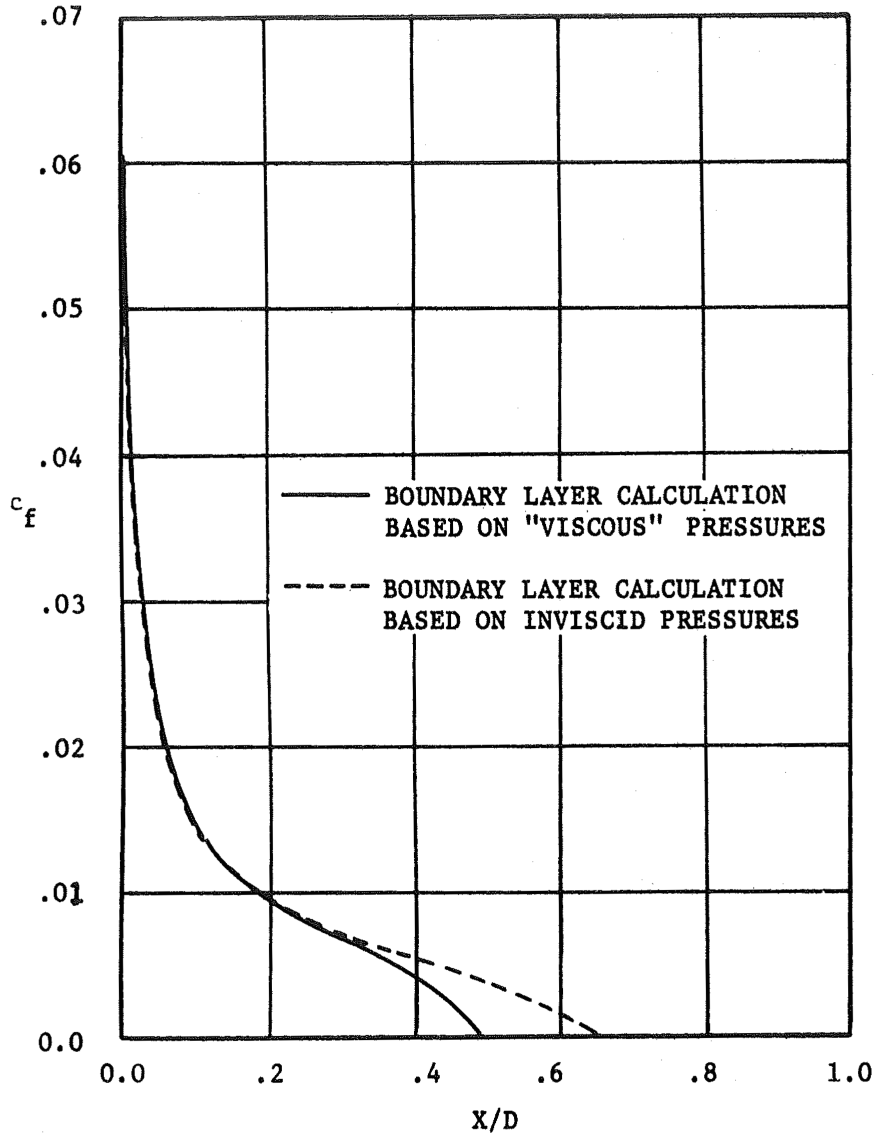


FIGURE . EFFECT OF "VISCOUS" MODELING ON CALCULATION OF LOCAL SKIN FRICTION COEFFICIENT FOR SPHERE IN SUBCRITICAL REGIME

✠ U.S. GOVERNMENT PRINTING OFFICE: 1975—635-049 / 69

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958